

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Hyvinvointiteknologia

2015

Miriam Reiman

BUSINESS INTELLIGENCE -RAPORTOINNIN KEHITYS



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Miriam Reiman

BUSINESS INTELLIGENCE -RAPORTOINNIN KEHITYS

Tämä projekti on toimeksiantajana toimivan yrityksen sisäistä kehitystyötä. Projektin tarkoituksena on parantaa yrityksen kustannustenseurantaa ja tuoda arvokas tieto helposti nähtäville, Business Intelligence -raportoinnin avulla ja helpottaa yrityksen johtamiseen liittyvien päätöksen tekemistä.

Raportointi toteutettiin SAP BusinessObjects -välineillä, yrityksen kirjanpitoaineistosta. Raportointi mahdollisti aineiston tarkastelun kokonaisuutena sekä sopimuksittain, palveluittain ja asiakkuuksittain. Raporteilla on lisäksi erilaisia suodatusvaihtoehtoja, joilla aineistoa voi katsella monesta eri näkökulmasta.

Raportointia varten luotiin tietokannat ja tehtiin tarvittava ajologiikka, lähdetiedostojen käsittelemiseksi, jotta tarvittava tieto saatiin haluttuun muotoon. Tietokannan päälle luotiin SAPin Universe -kerros, jolloin loppukäyttäjän ei tarvitse tuntea tietokantojen rakennetta tehdäkseen kyselyitä raporteilla. Näiden jälkeen luotiin raportointi, johon sisältyi kolme erillistä raporttia.

Aluksi työssä esitellään liiketoimintatiedon hallintaa (Business Intelligence, BI) ja sen työvälineitä. Tässä työssä käytetty BI-väline on SAP BusinessObjects. Työssä perehdytään lyhyesti myös tietovaraston rakenteeseen ja tehtäviin.

Projektin vaiheet koostuivat suunnittelusta, toteutuksesta ja testauksesta. Toteutus sisältää tietokantojen luomisen, ETL-kehityksen (Extract – Transform – Load) ja Universe -kerroksen luomisen tietovaraston ja loppukäyttäjän väliin, sekä itse raportin kehittämisen. Tämän lisäksi projektiin valmistuttua siirryttiin kehitysympäristöstä tuotantoympäristöön.

Tiukasta aikataulusta huolimatta projekti saatiin valmiiksi aikataulussa ja raportointi palvelee nyt loppukäyttäjää. Raportoinnin kehitys jatkuu, kasvattamalla raportoinnin tarjoamaa tiedon määrää, jolloin saadaan laajennettua sen käyttäjäkuntaa ja raportoinnista saataisiin irti kaikki mahdollinen hyöty.

ASIASANAT:

Business Intelligence, ETL, Tietokanta, Web Intelligence

Miriam Reiman

DEVELOPMENT OF BUSINESS INTELLIGENCE REPORTING SYSTEM

This thesis deals with the internal development of the client organization. The main purpose of this project was to enhance the client organization's ability to track its accounting data by making all the valuable information easily available using Business Intelligence.

Reporting on the organization's data is made using SAP BusinessObjects development tools, which enable users to analyze the data as a whole or by contract, services, or customership. Reports also include different sorting options, therefore it is possible to analyze the data according to these options.

Before the reporting system could be created, it was necessary to create the databases and ETL (Extract – Transform – Load) logic which would convert the data into a clean and usable form. SAP Web Intelligence reporting also needs a Universe layer between the databases and the reporting portal, so that users do not have to know the structure of the database to make queries. Universe layer is read-only so the end users cannot accidentally change values in the database. Reporting system is compiled from three different reports.

The phases of the project were planning, implementation and testing. In the planning phase, the requirements and needs were specified in a way that project goal was clear. The databases, the ETL logic, the Universe layer and the reporting system were created during the implementation phase. Testing was carried out throughout the whole project.

Despite the strict schedule, the project was completed on time and the end users could extract reports. The development of the reporting system still continues in the organization. The goal is to increase the information, expand the end user group, and gain the maximum benefits out of the reporting system.

KEYWORDS:

Business Intelligence, Database, ETL, Web Intelligence

SISÄLTÖ

KÄYTETYT LYHENTEET JA SANASTO	6
1 JOHDANTO	7
2 BUSINESS INTELLIGENCE JA TIETOVARASTO	8
2.1 Business Intelligence	8
2.1.1 ETL-kehityksen teoria luku	9
2.1.2 SAP BusinessObjects	10
2.2 Tietovarasto	11
2.2.1 Staging Area	11
2.2.2 Data Warehouse	12
2.2.3 Data Mart	12
3 PROJEKTI	14
3.1 Suunnittelu ja aikataulu	14
3.2 Toteutus ja työvälineet	15
3.3 Testaus	18
4 TIETOKANNAT JA AJOLOGIIKKA	20
4.1 Lähdetiedostot	21
4.2 SA-taso	25
4.2.1 SA-tason tietokanta	26
4.2.2 SA-tason ajologiikka	27
4.3 DM-taso	29
4.3.1 DM-tason tietokanta	30
4.3.2 DM-tason Ajologiikka	32
5 UNIVERSE JA RAPORTOINTI	36
5.1 Universe	36
5.2 Web Intelligence-raportointi	38
6 YHTEENVETO	41
LÄHTEET	42

KUVAT

Kuva 1. Tietovaraston arkkitehtuuri. Radu, Opera ja Fina toimivat lähdejärjestelminä kuvassa. [3]	11
Kuva 2. SA-tason latauksen kulkukaavio	17
Kuva 3. DM-tason latauksen kulkukaavio	17
Kuva 4. Koko prosessin vaiheiden kulkukaavio	18
Kuva 5. Rajapintakuvaksen malli.	20
Kuva 6. Tuotteiden ja palveluiden koodisto	22
Kuva 7. Sopimus koodisto	22
Kuva 8. Koodisto – Kustannuspaikka	23
Kuva 9. Varsinainen yrityksen käytössä oleva asiakaskoodisto	23
Kuva 10. Toisen ohjelman asiakaskoodisto, joka yhdistettiin varsinaiseen asiakaskoodistoon.	23
Kuva 11. Malli pääkirjasta.	24
Kuva 12. Malli virhe-Excelistä	24
Kuva 13. SA-tason latauksen kokonaisuus	25
Kuva 14. ER-studiolla tehty mallinnus SA-tasolta, johon on lisätty mallinnuksen vaihejako.	26
Kuva 15. Lähdetiedostojen tietokantaan tuonti	27
Kuva 16. Tapahtumien yksilöllinen numerointi ja tilikartan luonti	28
Kuva 17. Kommentosarja joka tarkistaa löytyykö tiedostoa nimeltä pääkirja	29
Kuva 18. DM-tason latauksen kokonaisuuden yleisnäkymä	30
Kuva 19. ER-studiolla mallinnetut DM-tason taulut	31
Kuva 20. Esimerkki Lippu -kentistä	31
Kuva 21. Korjatun virhe-Excelin lataaminen tietokantaan	32
Kuva 22. Pääkirjan laskentasäännöt ja poikkeuskäsittelyt	33
Kuva 23. Koodistojen tuonti SA-tasolta DM-tasolle	33
Kuva 24. Insert- ja Update-objektit sekä Key Generator	34
Kuva 25. Kommentosarja tiedoston lukemisen jälkeen arkistoon siirtoa varten	34
Kuva 26. Faktataulun lataus	35
Kuva 27. Kommentosarja joka tyhjentää DM-tason faktataulun	35
Kuva 28. Universe-kerroksen kaavio, dimensio- ja faktataulujen välisen linkityksen luominen	36
Kuva 29. Mittarin luominen Universe -kerrokseen	37
Kuva 30. Web Intelligence-raportin kaikki tapahtumat	38
Kuva 31. Web Intelligence-raportti, sopimukset välilehti	39
Kuva 32. Web Intelligence-raportin kyselyn luominen	40

KÄYTETYT LYHENTEET JA SANASTO

BODS	SAP BusinessObjects Data Services
BI	Business Intelligence, liiketoimintatiedon hallinta.
CSV	Comma-Separated Values, tiedostomuoto, jolla tallennetaan yksinkertaista taulukkomuotoista tietoa tekstitiedostoon.
DM	Data Mart, Datamartti
DW	Data Warehouse, Tietovarasto, tässä työssä käytetään lyhennettä DW, kun puhutaan tietovaraston DW-kerroksesta.
ETL	Extract – Transform – Load, Poiminta – Muokkaus – Lataus, lyhennettä käytetään kun puhutaan työssä tehdyistä latauksista ja ajologiikasta.
NULL	Tuntematon tai puuttuva arvo, ei kuitenkaan tarkoita tyhjää (välilyöntiä) tai nollaa.
Operatiivinen tietokanta	Perusjärjestelmän (esim. potilasjärjestelmän) tietokanta
Repository	Tietohakemisto
SA	Staging Area, tietokannassa oleva lastaus- tai työalue.
SAP BI	SAP Business Intelligence
SAP BO	SAP BusinessObjects
SQL	Structured Query Language, standardoitu kyselykieli.
Tietovarasto	Yleiskäsite tietovarastolle, myös Data Warehouse, puhuttaessa tietovarastosta tässä työssä tarkoitetaan tietovarastoa kokonaisuutena.
Universe	Semanttinen kerros, joka sijaitsee tietovaraston ja loppukäyttäjän välillä. Kuuluu SAP BO tuoteperheeseen.
XLS	Taulukkomuotoisen tiedoston yleismuoto.

1 JOHDANTO

Nykymaailmassa kaikenlainen tieto yritysten liiketoiminnasta on kasvattanut jatkuvasti arvoaan. Tämä projekti on Medbit Oy:n sisäistä kehitystyötä, yrityksen kustannustenseurannan parantamiseksi, Web Intelligence-raportoinnilla. Kustannusten tehokkaampi seuraaminen helpottaa yrityksen johtamiseen liittyvien päätösten tekoa sekä selkeyttää ja ohjaa liiketoimintaa.

Tämän projektin toimeksiantajana toimii Medbit Oy, joka tuottaa tietojärjestelmien käyttö-, kehittämis- ja ylläpitopalveluja terveydenhuollon ja sosiaalihuollon organisaatioille, pääasiassa Varsinais-Suomen ja Satakunnan alueilla.

Kustannustenseuranta tuotettiin SAP BusinessObjects välineillä, joilla Web Intelligence-raportointi toteutettiin. Raportoinnin avulla yrityksen kustannuksia pystytään seuraamaan kokonaisuutena, sekä lisäksi sopimuksittain, palveluittain ja asiakkuuksittain. Tämä projekti toteutettiin, koska yrityksen tilitoimiston ohjelmistot eivät kyenneet tekemään tarvittavia raportteja, joita yrityksen johto tarvitsi tuloslaskelman tukena, tulevaisuuden kustannusneuvotteluissaan.

Työssä tutustutaan Business Intelligenceen ja sen sisältämiin työvaiheisiin. Lisäksi tutustutaan tietovaraston rakenteeseen ja sen eri kerroksien tehtäviin ja rakenteisiin. Tässä työssä sanalla Tietovarasto viitataan tietovarastoon kokonaisuutena ja lyhenteellä DW (joka on suomeksi myös tietovarasto) viitataan tietovaraston DW-kerrokseen.

Projektin aikana tuotettiin vaatimusmäärittelyä vastaava raportointi. Projektin vaiheisiin kuuluivat suunnittelu, aikataulut, toteutus ja testaus. Vaatimusmäärittely aloitettiin käymällä läpi lähdetiedostoja ja tekemällä niihin tarvittavia lisäyksiä. Tämän jälkeen suunniteltiin tietokannat, rakennettiin ajologiikka (ETL-lataukset) ja luotiin Universe-kerros, tietovaraston ja loppukäyttäjän välille. Viimeisenä tehtiin määritelty raportointi.

2 BUSINESS INTELLIGENCE JA TIETOVARASTO

2.1 Business Intelligence

Business Intelligence (BI, liiketoimintatiedon hallinta), tarkoittaa yleisesti yrityksissä olevaa teknistä tiedon jalostus-, ja analysointiprosessia sekä raportointia. Vaikka tietoa tulee paljon nykyaikana eikä sen tallentaminen ole enää ongelma, niin kaiken sen tietomäärän hyödyntämien on monille yrityksille haaste. Tämän takia erilaisiin BI-ratkaisuihin panostetaan koko ajan enemmän, tarkoituksena auttaa yrityksen johtoa ja muita loppukäyttäjiä tekemään päätöksiä nopealla tahdilla varmemmin ja tehokkaammin helposti saatavan ja analysoitavan tiedon perusteella. Yksi BI:n tärkeimmistä ominaisuuksista on raportointi, joka mahdollistaa automatisoidut reaaliaikaiset raportit sekä käyttäjäkohtaiset kyselyt, jolloin jokaisen on mahdollista hakea tarvitsemansa tieto helposti ja nopeasti itse.[1, 2]

Business Intelligencen perustana ja lähteenä toimivat usein erilaiset tietovarastoratkaisut. Tietovarastoissa lähdejärjestelmistä kerätään tiedot yhteen sellaiseen muotoon, josta erilaisten BI-välineiden on mahdollista tehdä kevyitä kyselyitä, eikä lähdejärjestelmiä tarvitse kuormittaa turhaan. [1, 2]

BI-ohjelmistoista tarjontaa löytyy nykypäivänä todella paljon ja yhdelle yritykselle paras vaihtoehto ei välttämättä taas sovi toisen yrityksen tarpeisiin. [1, 3]

Muutamia tunnetuimpia Business Intelligence-välineitä:

- SAP BusinessObjects
- Microsoft SSIS ja SSRS (SQL reporting services)
- IBM Cognos
- Oracle data integrator
- QlikView
- SAS.

Medbitissä pääsääntöisesti käytössä oleva BI-väline on SAP BusinessObjects, mutta käytössä on muitakin, kuten Microsoftin SSRS (SQL Reporting Services), IBM Cognos kuutiot sekä SAS. [3]

2.1.1 ETL-kehityksen teoria luku

ETL (Extract – Transform – Load, eli poiminta – muokkaus – lataus) on tietovarastoinnin prosessi, jossa data haetaan operatiivisesta järjestelmästä, muokataan haluttuun muotoon ja siirretään tietovarastoon. Poiminta voi tapahtua suoraan operatiivisesta tietokannasta (vetomenetelmä), jolloin pitää huomioida kyselyiden vaikutus operatiivisen järjestelmän toimintaan. Kyselyn pitäisi olla tarpeeksi kevyt, jotta se ei kuormita ja hidasta lähdejärjestelmää turhaan, tai sitten poiminta voi tapahtua siirtotiedoston kautta (työntömenetelmä), jolloin operatiivinen järjestelmä antaa ulos esimerkiksi Excel-tiedostona tarvittavan datan. [1, 4, 5]

ETL-latauksen muokkaus tapahtuu pääasiassa tietovaraston SA-tasolla, joka toimii lastauslaiturina sisään tulevalle datalle. Muokkaus vaiheessa tietoja käsitellään niin, että niistä saadaan yhdenmukaisia eri lähdejärjestelmien välillä. Dataa muokataan myös usein helpommin luettavampaan muotoon, esimerkiksi yhdistämällä koodiperäisiin tietoihin niiden sanallisia selitteitä ja tarvittaessa luodaan myös rivien yksilöivät avaimet jos sellaisia ei tule lähdejärjestelmästä valmiiksi. Muokkaus vaiheessa tarkistetaan myös tietojen oikeellisuutta, etsimällä turhia tuplarivejä sekä puutteellisia ja virheellisiä rivejä, jotka joko korjataan tai niistä raportoidaan eteenpäin korjauksesta vastuussa olevalle. [1]

Poiminnan ja muokkauksen jälkeen tulee latausvaihe, jossa tiedot ladataan tietovaraston seuraavaan kerrokseen, riippuen tietovaraston rakenteesta se voi olla DW- tai DM-kerros. ETL-latauksen tekemiseen käytetään usein jotain siihen tarkoitettua ohjelmistoa, jotka osaavat myös suorittaa kantaan kirjoituksen ja suoraan kannasta lukemisen, tässä työssä on käytetty SAP BusinessObjects Data Services -ohjelmistoa. [1]

2.1.2 SAP BusinessObjects

SAP on maailman kolmanneksi suurin, itsenäinen ohjelmiston tuottaja ja SAP BusinessObjects (SAP BO) on SAPin Business Intelligenceen erikoistunut laaja tuoteperhe, johon kuuluvat mm. seuraavat ohjelmat:

- SAP BusinessObjects Web Intelligence
- SAP BusinessObjects Desktop Intelligence 3.1
- SAP BusinessObjects BI platform Information Design Tool (IDT)
 - Tämä vastaa vanhan version Universe designeria
- SAP Lumira
- SAP BusinessObjects Dashboards
- SAP BusinessObjects Design Studio
- SAP Crystal Reports
- SAP BusinessObjects Analysis, edition for Microsoft Office
- SAP BusinessObjects Explorer
- SAP BusinessObjects Mobile. [6, 7, 8]

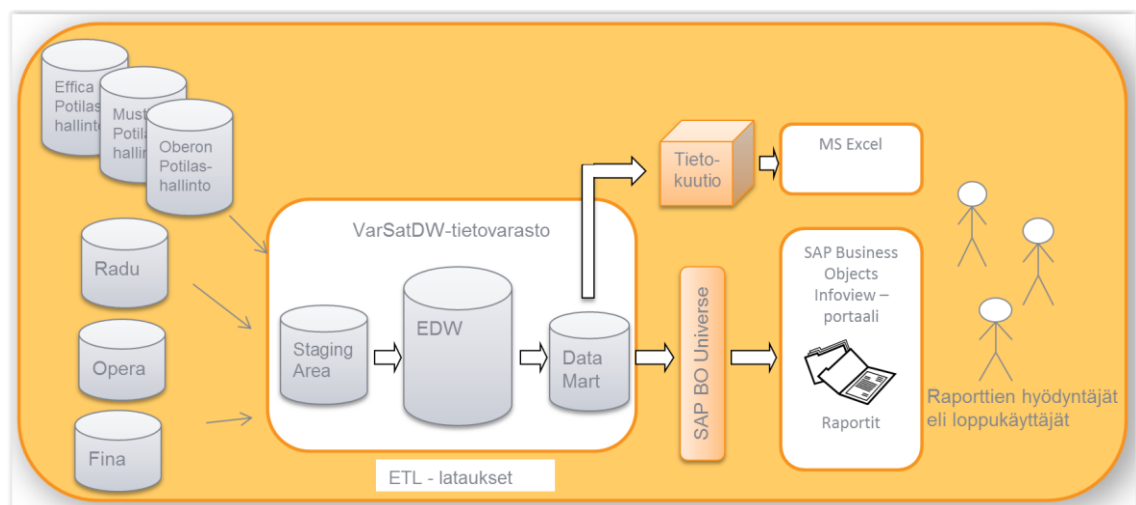
ETL-kehityksen tekemiseen tuoteperheestä löytyy visuaalinen SAP BO Data Services Designer-työkalu. Universe-kerroksen luomisen mahdollistaa SAP BO Universe designer. Universe rakennetaan tietovaraston päälle ja käyttävät esimerkiksi raportointi välineet SAP BO Web Intelligence ja Desktop Intelligence, millä tehdään raportointi loppukäyttäjille. Web Intelligence ja Desktop intelligence tarjoavat käyttäjilleen mahdollisuuden hakea tarvitsemiaan tietoja itse. [7]

Universe on semanttinen kerros, joka tulee tietovaraston päälle tietovaraston ja loppukäyttäjän väliin. Tässä kerroksessa tietovarastossa käytetyt termit käännetään loppukäyttäjille ymmärrettäväksi, jolloin heidän ei tarvitse ymmärtää tai tuntea tietovaraston rakennetta käyttäkseen siellä olevaa tietoa. Loppukäyttäjien ei tarvitse osata kirjoittaa universe -kerrokseen kohdistuvia kyselyitä SQL-kielellä ja koska universe on vain lukumuotoinen, loppukäyttäjät eivät pysty vahingossa muokkaamaan tai poistamaan tietovaraston tietosisältöä. [9, 10]

2.2 Tietovarasto

Organisaation tiedot ovat yritykselle arvokas resurssi, joita pitäisi kehittää aivan samoin kuin yrityksen muitakin resursseja, esimerkiksi henkilöstöä. Kuitenkin organisaation tiedot ovat usein hajallaan erilaisissa perusjärjestelmissä eli operatiivisissa tietokannoissa, joihin tiedot tulevat suoraan järjestelmien tapahtumista. Operatiiviset tietokannat eivät keskustele keskenään muiden järjestelmien kanssa, joten näissä oleva tieto on täysin eristyksissä järjestelmien sisällä, eikä se ole helposti tulkittavissa tai edes kaikkien sitä tarvitsevien saatavilla. [1]

Tietovaraston (Kuva 1) tehtävänä on tuoda nämä tiedot sitä tarvitseville helposti analysoitavaksi ja tulkittavaksi, sekä vastata monien käyttäjäryhmien erilaisiin tarpeisiin olemalla kokonaisuus, jonka eri tasoissa operatiivisten tietokantojen tieto integroidaan ja jalostetaan raportoitavaan muotoon. [1]



Kuva 1. Tietovaraston arkkitehtuuri. Radu, Opera ja Fina toimivat lähdejärjestelminä kuvassa. [3]

2.2.1 Staging Area

Staging Area (SA) tarkoittaa tietovaraston lastaus- tai työaluetta, joka sijaitsee lähdejärjestelmien ja tietovaraston muiden kerroksien, DW-tason tai Datamartin välissä (Kuva 1). Työaluetta käytetään eri lähdejärjestelmien datan kokoamispaikkana ja yhdenmukaistamiseen jatkokäsittelyä varten. Tietovaraston tältä

tasolta ei yleensä tehdä kyselyitä tai raportteja eikä loppukäyttäjille anneta oikeuksia tälle tasolle. [1]

SA-tasolle data voidaan tuoda sisään joko lukemalla se suoraan lähdejärjestelmän tietokannasta tai tulostamalla tiedot erilliselle siirtotiedostolle, esimerkiksi Excel-tiedostoksi. [1]

2.2.2 Data Warehouse

Tietovaraston rakenteessa (Kuva 1) Data Warehouse-kerros (DW) sijoittuu SA-tason jälkeen ja ennen DM-tasoa, jos sellainen on tietovarastomalliin suunniteltu. DW-kerroksessa tieto on mallinnettua, tarkantasoista tietoa, eli DW pitää sisällään isoja tieto kokonaisuuksia yhdessä paikassa, useista eri lähdejärjestelmistä. [1, 3, 11]

DW-kerros on lähinnä tietojen arkistointia ja historiointia varten, mutta siellä voidaan jatkaa tiedon laadunparannusta ja oikeellisuuden tarkistamista. Esimerkiksi tietynlaisia kirjoitusvirheitä voidaan korjata automatisoidusti. Tältä tasolta kyselyiden tekeminen on raskasta ja hidasta, joten raportointia varten luodaan usein erikseen datamartti. [3]

2.2.3 Data Mart

DM-taso, tuttavallisemmin datamartti, on osa tietovaraston kokonaisuutta, mikä tulee tietovaraston päällimmäiseksi kerrokseksi (Kuva 1), joissa jo SA-tasolla tai DW-kerroksessa oleva tietomassa jäsennetään järkeviksi ja pienemmiksi kokonaisuuksiksi. Tämä mahdollistaa kevyet kyselyt tietokannasta, mikä tukee hyvin raportointia. Datamartti vastaa siis jonkin tietyn käyttäjäryhmän tiettyihin vaatimuksiin. Suunnittelu alkaakin usein kyseisten käyttäjien tarpeiden määrittelyllä, jonka mukaan datamartin rakenne voidaan suunnitella palvelemaan käyttäjiään tehokkaasti. [1, 12, 13]

Ralph Kimballin mukaan datamartin suunnittelua kannattaisi lähestyä dimensiomallilla (kutsutaan myös tähtimalliksi), joka koostuu dimensio-, ja faktatauluista. Faktataulut sisältävät yksilölliset tapahtuman tarkasteltavat tiedot, kun taas dimensiotaulut koostuvat pysyvämmistä, useammassa tapahtumassa esiintyvistä tiedoista, jotka linkittyvät faktatauluun. Esimerkki tapauksena käsitellään vaikka yrityksen myyntejä, jolloin faktataulu sisältäisi tilauksen numeron, päivämäärän, tilattujen tuotteiden määrän ja hinnan, jotka ovat yksilölliset jokaisella tilauksella. Asiakkaiden perustiedoista ja tuoteluettelosta taas tulisi dimensiotauluja, koska yksi asiakas voi tehdä useampia tilauksia ja yksi tuote voi kuulua useampaan tilaukseen. Dimensiotauluja voidaan hyödyntää myös useammassa kuin yhdessä faktatauluissa ja niiden sisältämä tieto on normalisoitua, joka tarkoittaa tietojen toiston vähentämistä. [1, 14, 15]

3 PROJEKTI

Projektin tavoitteena oli tehdä Medbitin johdolle Web Intelligence-raportointi, joka mahdollistaisi organisaation kustannusten tarkemman seuraamisen sopimuksittain, asiakkuuksittain sekä palveluittain.

Raportilta tulee selvitä tietyt tiliöinti tiedot, jotka tulevat tilitoimistolta saatavasta pääkirjasta, sekä erillisestä yrityksen sisällä manuaalisesti ylläpidetystä Excel -koodistosta, joka sisältää asiakas-, sopimus-, palvelu- ja kustannuspaikka tiedot. Pääkirjasta tulevat tiedot ovat koodimuotoista yksittäiselle tapahtumalle kohdistuvaa tietoa, joihin kuuluvat tositenumero, päivämäärä, sopimusnumero, asiakasnumero, kustannuspaikka ja kustannukset: debet, kredit ja alv. Koodistosta täydennetään pääkirjan tietoja koodien sanallisilla arvoilla, kuten asiakkaan nimi, sopimuksen nimi, kustannuspaikan nimi, sekä sopimuksista johdetut maksajaryhmän tiedot (sopimuksen ylätaso ja maksajaryhmä). Muita tarvittavia tietoja ovat tapahtuman summa, joka lasketaan tilitoimiston laskusääntöjen mukaisesti, ja tositelaji sekä vuosi ja vuosikuukausi, jotka johdetaan ETL-latauksissa tapahtuman tositenumeroista ja päivämäärästä.

3.1 Suunnittelu ja aikataulu

Projektin suunnitteluvaiheessa lähdettiin tekemään vaatimusmäärittelyä, selvittämällä raportin todellinen tarve ja käyttötarkoitus. Projektin toimeksiantajalle esitettiin seuraavanlaisia kysymyksiä:

- mitä tietoja raportilta halutaan nähdä,
- mihin tarpeeseen tietojen tulee vastata,
- onko raportin tarve kertaluonteinen, kuukausittainen vai vuosittainen,
- minkälainen on tietokannan ja raportointiympäristön tulevaisuuden käyttö, pitääkö suunnitella jatkuvuutta.

Näiden kysymyksien avulla määriteltiin millaisia lähdetiedostoja projektissa tarvitaan, minkälainen tietokantarakenteen pitää olla, millainen universe voidaan rakentaa ja minkälaisia mittareita sinne pitää pystyä tekemään.

Vaatusmäärittelyn pohjalta pystyttiin suunnittelemaan projektin aikataulu, arvioimalla tulevat työtehtävät ja niihin kuluva aika. Rajapintakuvauksen tekeminen alkoi myös heti alussa vaatimusmäärittelyn jälkeen, jota korjailtiin ja täydennettiin projektin edetessä tarpeen mukaan. Rajapintakuvauksen tarkoituksena oli pitää kehittäjät ajan tasalla mukana olevien koodistojen, tietokannan, universe-kerroksen ja raportoinnin vastaavuuksista niin, että tietty lähteestä tuleva tieto voidaan helposti paikantaa mistä vaiheesta prosessia tahansa. Projektin aikana muita dokumentteja olivat projektin aikainen dokumentaatio, aikataulu, kokousmuistiot sekä kaaviot ja piirrokset.

Tehtävän toimeksianto tuli organisaation johdon kiireellisestä tarpeesta saada kustannusseuranta raportointi. Projekti toteutettiin hyvin nopealla aikataululla, koska toimeksianto tuli tietoon lokakuussa ja raportoinnin piti olla valmis ja käytettävissä joulukuun ensimmäisellä viikolla. Projektin kaikkien vaiheiden ei kuitenkaan tarvinnut olla valmis vielä tällöin, raportoinnin valmistumisen jälkeen jäljellä oli vielä hienosäätöä, ajojen parantamista ja raportoinnin ulkonäön parannuksia sekä dokumentaation tekemistä.

3.2 Toteutus ja työvälineet

Projektin toteutuksessa on ollut käytössä seuraavat työvälineet:

- SAP BO 3.1
- SAP BusinessObjects Data Services Designer 2009
- SAP BusinessObjects Infoview
- SAP BusinessObjects Universe designer
- Microsoft SQL server management studio 2014 versio 12.0.2
- ER-studio Data Architect 9.1.3 (Embarcadero)
- Microsoft Visio 2010
- Microsoft Office 2010

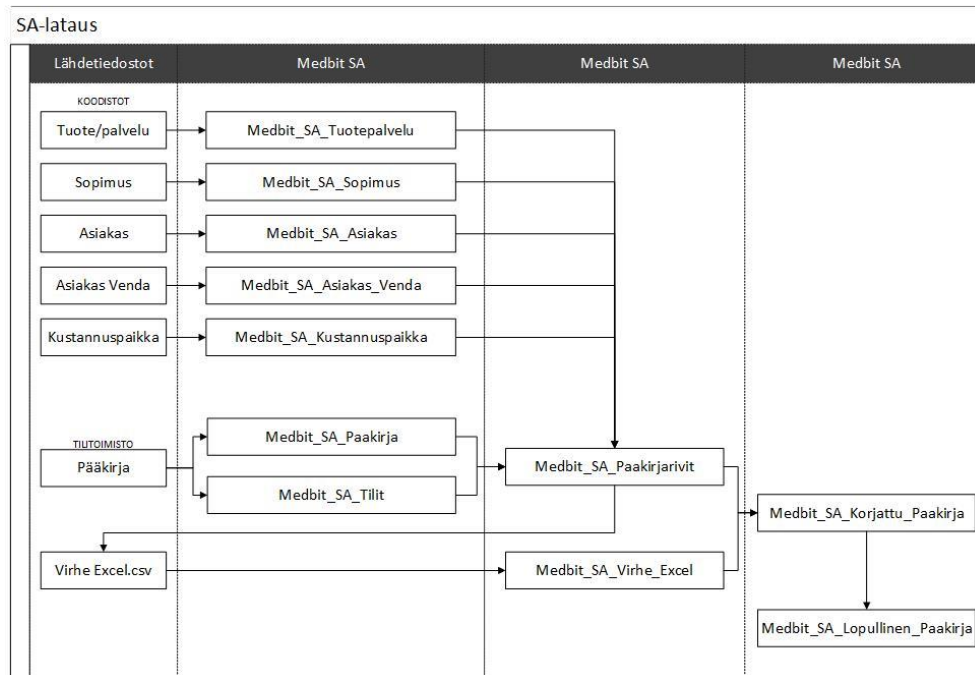
- Microsoft Word 2010
- Microsoft Excel 2010
- Microsoft PowerPoint 2010.

SAP BusinessObjects 3.1 -ohjelmista ETL-kehityksen työvälineenä, eli ajojen tekemiseen ja ajamiseen, oli SAP BO Data Services Designer. Universen suunnitteluun ja luomiseen käytettiin SAP BO Universe Designer -ohjelmaa ja raportoinnin tekemiseen käytettiin SAP BO Infoview -portaalia, jota myös loppukäyttäjät käyttävät raportoinnin katseluun ja osa myös kehittämiseen.

ER-studio Data Architectia käytettiin tietokantojen mallintamiseen ja luomiseen. Tietokantojen katseluun ja SQL-kyselyiden tekemiseen sekä aineiston testaukseen käytettiin Microsoft SQL-server management studio 2014.

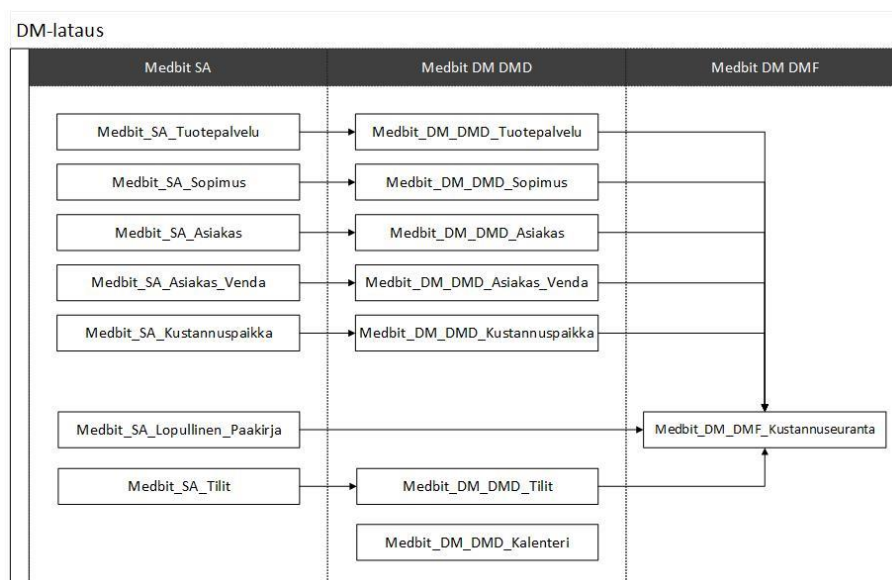
Projektin suunnitelmia ja dokumentaatioita tehtiin Microsoft Office 2010 välineillä MS Wordilla, PowerPointilla ja Excelillä. Aikataulu sekä korjauskierroksien kulkukaaviot suunniteltiin MS Visio 2010 -ohjelmalla. Lähdetiedostojen käsittely tapahtui MS Excelillä.

Microsoft Visiolla tehdyt projektin kulkukaaviot kuvaavat tietokantojen eri tasoissa tapahtuvan tiedonkulun sekä koko prosessin vaiheiden kiertokulun. Kuva 2 kertoo SA-tasolle ladattavien lähdetiedostojen siirtovaiheet lopullisiin tauluihin. Koodistot menevät sellaisinaan tietokantaan, mutta pääkirjalle tehdään muutamia muutoksia, jonka takia se käy läpi useamman vaiheen, ja siitä erotellaan tilikartta omaksi taulukseen.



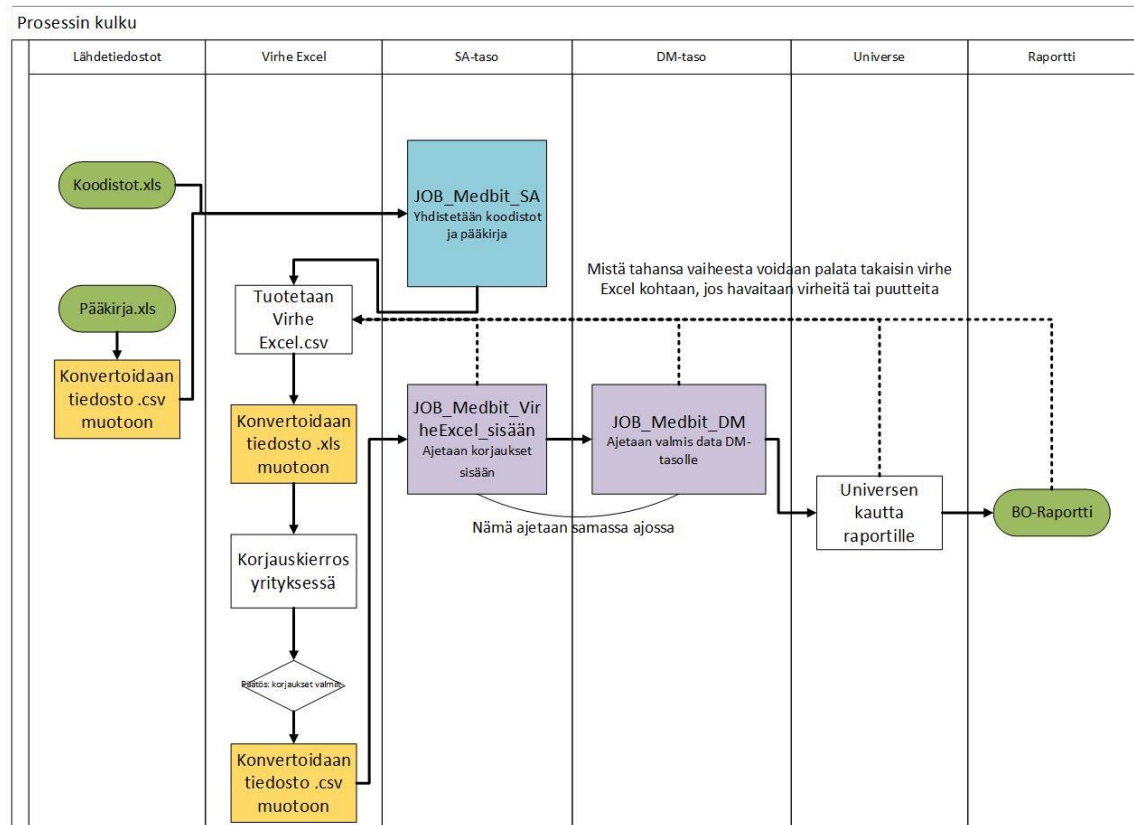
Kuva 2. SA-tason latauksen kulkukaavio

DM-tason kulkukaavio (Kuva 3.) kertoo, miten dimensio- ja faktataulut koostetaan SA-tason lähteistä. Faktataulu koostuu pääkirjasta johon on lisätty koodistojen tiedot ja DM-tasolle on lisätty vielä kalenteritaulu, jotta ajoon saadaan tarvittavat aikatiedot.



Kuva 3. DM-tason latauksen kulkukaavio

Kuvassa 4, oleva kaavio kertoo missä järjestyksessä prosessin eri vaiheet toteutetaan. Prosessi alkaa lähdetiedostojen lataamisesta tietokantaan, jonka jälkeen luodaan virhe Excelin, joka lähtee korjauskierrokselle. Virhe-Excelin lataamisen jälkeen tiedot menevät raportointiin.



Kuva 4. Koko prosessin vaiheiden kulkukaavio

3.3 Testaus

Projektiin aikana suoritettiin jatkuvaa testausta. Esimerkiksi ajoon tehtyjen muutoksien jälkeen ajo voitiin suorittaa, jolloin nähtiin menikö se onnistuneesti läpi vai ei. Ajon mennessä onnistuneesti läpi, tarkistettiin tietokannasta tietosisältö, jolloin nähtiin myös, että kaikki toimii niin kuin pitääkin, jos sisällöstä kuitenkin löytyi puutteita tai virheitä ongelmalla lähdettiin selvittämään ajamalla ajoa testitilassa, jolloin päästiin seuraamaan ajon jokaista vaihetta yksitellen ja näin voitiin paikantaa ongelmakohta latauksen sisältä.

Yleistä testausta on suoritettu niin, että tietokannassa olevia pääkirjan rivimääriä on verrattu lähteenä toimivan pääkirjan riveihin. Tietokannassa olevia summia verrattiin tuloslaskelman lukuihin, jolloin saatiin varmistus, että kaikki rivit ovat pysyneet mukana, eivätkä summat ole muuttuneet ajojen tai korjauksien aikana.

4 TIETOKANNAT JA AJOLOGIIKKA

Lähdetiedostojen saaminen analysoitavaan muotoon raportille vaatii sen, että tiedot näistä tiedostoista siirretään ensin tietokantaan, jossa niitä voidaan jatkojalostaa sellaiseen muotoon kuin raportointi vaatii.

Projektia varten luotiin oma tietokanta kehitysympäristöön, josta se siirrettiin projektin valmistuttua tuotantoympäristöön. Tietokanta sisältää SA- ja DM-tason, jotka mahdollistavat raportoinnin, ja koska tässä aineistossa ei tarvita historiointia, DW-kerros jätettiin pois. Tietokantojen luominen aloitettiin mallintamalla taulut käyttäen ER-studiota ja taulujen kentät perustuvat lähdetiedostoihin, joista data tullaan lopulta ajamaan sisään tietokantaan. Mallintamisessa käytettiin apuna aikaisemmin luotua rajapintakuvausta (Kuva 5), johon määriteltiin tarvittavat taulut, niiden kentät, kenttien tietotyypit sekä tiedostojen sijainnit ja kenttien kuvaukset.

	A	B	C	D	E	F
1	Field Name	Data type	Allow nulls	Source File	Source File from	Description
2	pääkirja(paakirjarivit)			Paakirja_Tili.xls	Tilitoimisto	
3	tili	varchar(50)	yes	Paakirja_Tili.xls	Y:\Yritys\Kustannuslaskenta\Kesken	Tapahtuman tilinumero
4	tosite	varchar(50)	yes	Paakirja_Tili.xls	Y:\Yritys\Kustannuslaskenta\Kesken	Tapahtuman tositteen numero
5	pvm	date	yes	Paakirja_Tili.xls	Y:\Yritys\Kustannuslaskenta\Kesken	Tapahtuman päivämäärä
6	debet	decimal(12, 2)	yes	Paakirja_Tili.xls	Y:\Yritys\Kustannuslaskenta\Kesken	€
7	kredit	decimal(12, 2)	yes	Paakirja_Tili.xls	Y:\Yritys\Kustannuslaskenta\Kesken	€

Kuva 5. Rajapintakuvaksen malli.

Tämän projektin ETL-kehitys tehtiin käyttäen työntömenetelmää, jolloin operatiivisesta järjestelmästä tarpeellinen tieto saadaan erillisenä tiedostona. Tämän menetelmän valinta oli luonnollinen, koska ei ollut mahdollisuutta saada oikeuksia suoraan tilitoimiston tietokantaan, kuten vetomenetelmään olisi tarvinnut.

Yrityksessä ETL-työkaluna on käytössä SAP BusinessObjects Data Services XI. Projektia varten kehittäjille luotiin oma kehitystietohakemisto (repository), johon määriteltiin yhteys projektia varten luotuun kehityskantaan.

4.1 Lähdetiedostot

Lähdetiedostoja olivat tilitoimistolta tuleva pääkirja sekä organisaation omat koodistot, jotka sisältävät asiakas, sopimus, tuote sekä kustannuspaikka tietouden. Pääkirja oli lähdetiedostoista ainoa, jonka muotoon ei pystytty vaikuttamaan. Pääkirja saatiin sellaisenaan tilitoimiston järjestelmästä ja sen muoto aiheutti myös haasteita projektin aikana. Koodistot olivat yrityksessä manuaalisesti ylläpidettäviä. Tilanne oli ihanteellinen, koska ylläpitäjiä oli vain yksi ja tällöin kyseistä henkilöä voitiin ohjeistaa täydentämään koodistoa tarvittavilla tiedoilla sekä tekemään se ilman turhia tyhjiä sarakkeita tai rivejä.

Koodistot ovat yhdessä Excel-tiedostossa eri välilehdillä ja tiedoston pitää olla BODS-ohjelmistoversion rajoittamissa puitteissa muotoa Excel 97 – 2003 (xls). Koodistoihin tehtiin manuaalisesti lisäyksiä raportoinnin tehostamiseksi, jotta kaikki tarvittava tieto voitiin raportoida. Koodistojen automatisoitu sisäänluku aiheutti muutamia haasteita manuaalisesti ylläpidettävien koodistojen kohdalla, koska niiden rakenteen ja sisältömuodon pitää pysyä muuttumattomana, eli sarakkeiden järjestyksen ja määrän pitää olla vakio, mutta rivien määrä saa vaihdella. Jommankumman muuttuessa todennäköisin seuraus olisi latauksen kaatuminen. Kaatuminen on kuitenkin helposti havaittavissa oleva virhe, mutta muutokset voivat aiheuttaa myös väärän tietosisällön raportille, mikä ei enää ole niin helposti havaittavissa oleva virhe.

Kuvat 6 – 10. kertovat koodistojen rakenteen ja esimerkin tietosisällöstä. Koodistoissa ensimmäisenä olevaa tyyppisaraketta ei tuoda sisään tietokantaan ja ensimmäisillä riveillä näkyvät vyörytyskoodit lisättiin tilitoimiston laskentasääntöjen takia, jotta raportoinnin vaatimukset saatiin täytettyä.

Kuva 6. Palveluun lisättiin järjestelmäluokka. Järjestelmäluokitus koostuu kirjaimista E, P ja Y, jotka määrittelevät millaisesta palvelusta tai tuotteesta on kyse. Tämän luokituksen avulla pystytään seuraamaan tarkemmin palveluiden laskuttamista.

	A	B	C	D
1	Tyyppi	Numer	Nimi	Jarjestelma_luok
2				
3	Tuote/palvelu	1999	Vyörytetyt kustannukset	Y
4	Tuote/palvelu	2001	Acute Työterveys	E
5	Tuote/palvelu	2002	AivoDiet	E
6	Tuote/palvelu	2003	Altti	P
7	Tuote/palvelu	2004	Alueellinen valtuutustenhallinta- ja potilastiedon luovutuspalvelu	P
8	Tuote/palvelu	2005	Arkkitehtuuripalvelu	Y
9	Tuote/palvelu	2006	Aromi	Y
10	Tuote/palvelu	2007	ARTTURI	Y

Kuva 6. Tuotteiden ja palveluiden koodisto

Kuva 7. Sopimuksen tietoihin lisättiin useampia lisätietoja sopimusrakenteen vuoksi. Sopimuksen nimessä näkyy sen yksilöivä Medbit-n-n -muotoa oleva yhdistelmä, joka oli raportoinnin kannalta tarpeellista erottaa sopimuksen nimestä. Erotetulle osalle luotiin oma kenttä sopimuksen_ID. Sopimuksien perusteella pitää päästä seuraamaan mille kunnille tapahtumia kohdistuu sekä mille maksajaryhmälle kustannuksia kohdistuu. Tämän takia sopimuksille määriteltiin ylätasen numero ja nimi, jotka kertovat kuntatiedot. Maksajaryhmä kertoo sopimuksen sairaanhoitopiiriin. Kustannuspaikka koodistoon (kuva 8.) ei tehty muutoksia.

	A	B	C	D	E	F	
1	Tyyppi	Numero	Nimi	sopimuksen_ID	ylataso_nro	ylataso_nimi	maksaja_ryhma
246	Sopimus	1253	MEDBIT-1-253-2014 Effector laskurivien käsittelymoduuli	MEDBIT-1-253-2014	1000		
247	Sopimus	2000	KUNTA1	2000 KUNTA1	2000	KUNTA1	
248	Sopimus	2001	Medbit-2-1 RTG kuvien pitkäaikaisarkistointi	Medbit-2-1	2000	KUNTA1	
249	Sopimus	2002	Medbit-2-2 Alueellinen silmäpohjakuvauspalvelu	Medbit-2-2	2000	KUNTA1	
254	Sopimus	3000	KUNTA2	3000 KUNTA2	3000	KUNTA2	
255	Sopimus	3001	Medbit-3-1 RISPCS-järjestelmä	Medbit-3-1	3000	KUNTA2	
280	Sopimus	5000		5000	5000		

Kuva 7. Sopimus koodisto

	A	B	
1	Tyyppi	Numero	Nimi
2			
3	Kustannus:10		Asiakaspalvelu
4	Kustannus:20		Sovelluspalvelu
5	Kustannus:30		Kehittämispalvelu
6	Kustannus:40		Käyttöpalvelut
7	Kustannus:50		Hallinto
8	Kustannus:60		Palvelu- ja projektimyynti
9	Kustannus:99		Yleiset
10			

TuotePalvelu Sopimus KP Asiakkaat Asiakkaat_Venda

Kuva 8. Koodisto – Kustannuspaikka

Asiakaskoodisto (Kuvat 9. ja 10.) oli kirjanpitojärjestelmän takia hieman haasteellinen, koska saman järjestelmän sisällä oli kaksi erillistä ohjelmaa osto- ja myyntilaskujen käsittelyyn. Kummallakin on omat asiakas koodistot, eivätkä ne keskustele keskenään, joten raportoinnin näkökulmasta samalla asiakkaalla voi olla kaksi eri asiakasnumeroa. Asiakaskoodistoon lisättiin kyseisen asiakkaan Venda-järjestelmässä oleva numero, ja tällä tavalla asiakkaan esiintyessä lähdeaineistossa kummalla tahansa numerolla. Näin asiakas pystyttiin tunnistamaan ja nimeämään oikein.

	A	B	C	D	
1	Tyyppi	Numero	Nimi	Numero Venda	
2	Asiakkaat	2999	Vyörytetty		
3	Asiakkaat	3001	Asiakas 1	21024	
4	Asiakkaat	3002	Asiakas 2	21065	
5	Asiakkaat	3003	Asiakas 3	21167	
6	Asiakkaat	3004	Asiakas 4	21136	
7	Asiakkaat	3005	Asiakas 5	21128	
8	Asiakkaat	3006	Asiakas 6	21033	

TuotePalvelu Sopimus KP Asiakkaat Asiakkaat_Venda

Kuva 9. Varsinainen yrityksen käytössä oleva asiakaskoodisto

	A	B	C
1	Tyyppi	Koodi	Nimi
2	Asiakkaat	19999	Asiakas 1
3	Asiakkaat	20001	Asiakas 2
4	Asiakkaat	20002	Asiakas 3
5	Asiakkaat	20201	Asiakas 4
6	Asiakkaat	21001	Asiakas 5
7	Asiakkaat	21002	Asiakas 6
8	Asiakkaat	21003	Asiakas 7

TuotePalvelu Sopimus KP Asiakkaat Asiakkaat_Venda

Kuva 10. Toisen ohjelman asiakaskoodisto, joka yhdistettiin varsinaiseen asiakaskoodistoon.

Pääkirja (Kuva 11.) tulee tilitoimistolta Excel-tiedostona, muodossa Excel 97 – 2003. Excelin omat muotoilusäännöt vaikeuttivat datan lukua ja virheraportointia, koska pääkirjalla samassa sarakkeessa voi olla useita eri tietotyyppisiä ja ositenumero on sellaisessa muodossa jonka Excel tulkitsee päivämääräksi. Lähtökohtaisesti lähdetiedoston sisältöä ei haluttu joutua manuaalisesti korjaamaan, joten ongelma ratkaistiin tallentamalla pääkirja tekstitiedostoksi (csv-luetteloerotin -muotoon), kun se saadaan tilitoimistolta. ETL-latauksien yhteydessä pääkirjalta poistettiin tyhjät rivit, eroteltiin tilinumerot omaan tietokantatauluun ja kullekin tapahtumariville lisättiin tilinumero. Alkuperäisessä pääkirjassa kukin tilinumero esiintyy vain kerran, kyseiseen tiliin kohdistuvien tapahtumien otsikkona.

TILI	TOSITE	PVM			DEBET	KREDIT	ALV	Osasto	Sopimus	Tuote/palvelu	Asiakas	SELITE
3020	Myynti 24%Muut palvelut											
			11-353	13.8.2014		-20	-3,87	99	21172	2156	99000	
			11-356	15.8.2014		-20	-3,87	99	21172	2156	99000	
			11-357	15.8.2014		-20	-3,87	99	21172	2156	99000	
			11-358	15.8.2014		-10	-1,94	99	21172	2156	99000	
			11-360	15.8.2014		-20	-3,87	99	21172	2156	99000	
			11-361	15.8.2014		-20	-3,87	99	21172	2156	99000	

Kuva 11. Malli pääkirjasta.

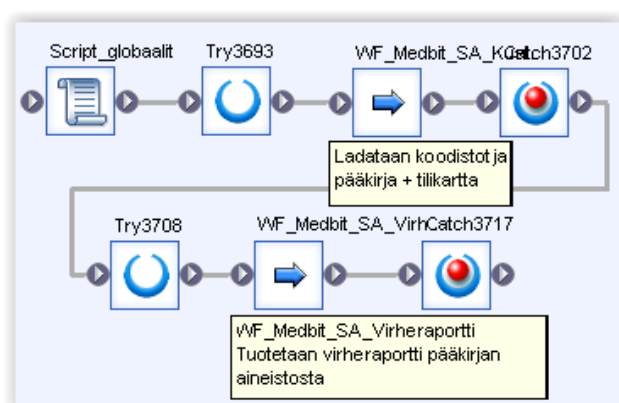
Yhtenä lähteenä toimii myös ajon prosessina syntyvä virhe-Excel (Kuva 12.), joka luodaan SA-tasolla ja luetaan sisään DM-tasolle. Virhe-Exceliin tulostetaan kaikki pääkirjalla olevat puutteelliset tai virheelliset rivit, esimerkiksi sellainen rivi jolta puuttuu asiakas tai vaikka sopimus tieto. Tiedosto kulkee yrityksen sisällä korjauskierroksen ennen tietokantaan takaisin viemistä.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	TOSITE	PVM	DEBET	KREDIT	ALV	SOPIMUS	SOPIMUKI	TUOTE/PATUOTE/PALVELUN NIMI	ASIAKAS	ASIAKKAAN NIMI	ASIAKAS \ ASIAKKAAN	OSASTO	OSASTON NIMI	TILI	TILIN NIM	NRO		
1	65-12957	1.12.2014				1138	Medbit-1-	2135 Tallennuskapasiteetti	20002						30401	Myynti 24	372	
2	40-12	31.12.2014				17004	Medbit-1-	2018 Carestream	21019			10	Asiakaspalvelu	30401	Myynti 24	332		
3	40-79	16.12.2014				21172	Medbit-2-	2156 VRK-korttipalvelu	99000			30	Kehittämispalvelu	3030	VRK Kortti	16		
4	40-12	31.12.2014				1007	Medbit-1-	2160 OSTI	3097			20	Sovelluspalvelu	4450	Atk-ylläpi	1300		
5	55-42636	4.12.2014				1235	MEDBIT-1-	2205 Safir-tuoteperhe				60	Palvelu- ja projektimyynti	4450	Atk-ylläpi	1410		
6	55-42783	16.12.2014				1160	Medbit-1-160-2013	Videonuottelu				60	Palvelu- ja projektimyynti	4450	Atk-ylläpi	1430		
7	90-385	30.12.2014						2030 Efecte				10	Asiakaspalvelu	4450	Atk-ylläpi	1549		

Kuva 12. Malli virhe-Excelistä

4.2 SA-taso

SA-tason latauksessa (Kuva 13.) tapahtuu lähdetiedostojen eli tilitoimiston pääkirjan ja yrityksen omien koodistoiden lukeminen tietokantaan. Pääkirjan koodiarvot yhdistetään koodistoissa oleviin vastaaviin selityksiin, jotta tietoja on helpompi tulkita. Tietokantataulun lisäksi, tässä latauksessa muodostetaan erillinen Excel tiedosto, virhe-Excel, joka tarkastetaan ja korjataan manuaalisesti organisaation sisällä.



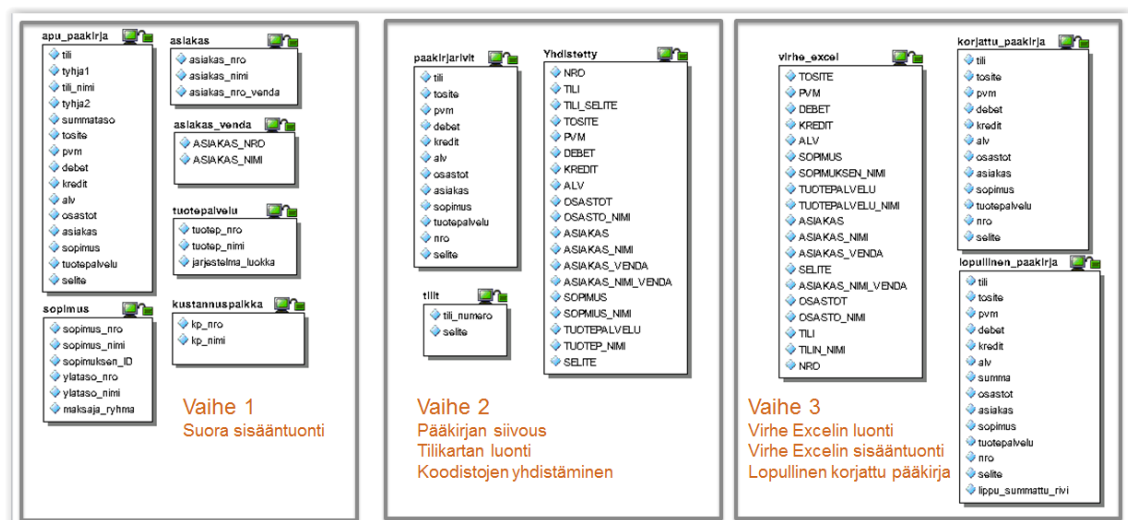
Kuva 13. SA-tason latauksen kokonaisuus

Try ja Catch-objektit (Kuva 13.) ovat aina pari. Jos ajossa tapahtuu virhe tällaisen parin sisällä, se tapahtuu hallitusti, eli catch-objekti kertoo mitä ajon kaatuessa tapahtuu. Tässä ajossa oleva catch-objekti sisältää komentosarjan, joka lähettää sähköpostin määriteltyyn sähköpostiosoitteeseen ja tiedottaa ajon kaatumisesta ja kaatumisen yhteydessä tulleesta virheilmoituksesta. Samassa kuvassa näkyvä WF_Medbit.. on WorkFlow, jonka sisälle tulee yksi tai useampi DataFlow, joiden sisälle varsinainen ajologiikka rakennetaan. Objektit workFlow ja DataFlow pitävät ajon hallinnassa ja näiden avulla määritellään missä järjestyksessä mikäkin ajon vaihe tapahtuu. Kun objektit on yhdistetty toisiinsa, kuten kuvassa 13. ajo suorittaa ne peräkkäin, siinä järjestyksessä kuin ne on laitettu, mutta jos objekteja ei yhdistettäisi toisiinsa, ajo suorittaisi nämä rinnakkain.

4.2.1 SA-tason tietokanta

SA-taso toimii lähdetiedostojen lastausalueena ja työalueena, tarkoittaen sitä, että SA-tasolla olevat taulut ja kentät suunniteltiin vastaamaan lähteinä olevien pääkirjan ja koodistojen rakenteita. SA-tasolla toteutettiin myös tarvittavan virhe-Excelin tekeminen, joissa yhdistetään pääkirjan ja koodistojen sisältämiä tietoja, joten SA-tason mallinnuksen kehitys jatkui samalla, kun SA-tason ETL-ajoa kehitettiin.

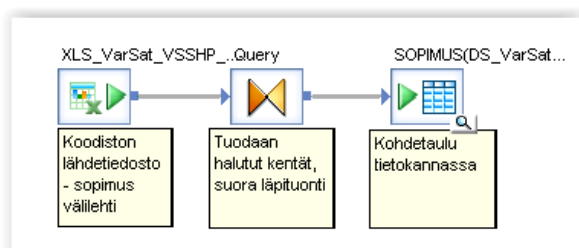
SA-tason taulut mallinnettiin (Kuva 14.) vaiheittain projektin edetessä. Ensimmäisessä vaiheessa luotiin taulut, jotka vastasivat suoraan sisään tuotavan aineiston kenttiä. Esimerkiksi apu_paakirjatauluun tuli kentät nimeltä tyhja1 ja tyhja2, jotka ovat oikeasti tyhjiä lähdetiedostossakin. Kentät luotiin pääkirjan rakenteen vuoksi, jotta tiedoston tietokantaan luku onnistui. Seuraavassa vaiheessa pääkirjalle luotiin uusi taulu, johon se tulee siivoamisen jälkeen. Pääkirjalta eroteltava tilitaulu luotiin tässä vaiheessa ja virhe-Exceliä varten tarvittava yhdistetty-tili, jossa pääkirjan koodiarvoihin yhdistettiin selitteet koodistoista. Kolmannessa vaiheessa luotiin taulut varsinaiselle virhe-Excelille, korjatulle pääkirjalle ja vielä tarpeellisten muutosten (joista kerrotaan tarkemmin seuraavassa luvussa), jälkeen lopulliselle pääkirjalle.



Kuva 14. ER-studiolla tehty mallinnus SA-tasolta, johon on lisätty mallinnuksen vaihejako.

4.2.2 SA-tason ajologiikka

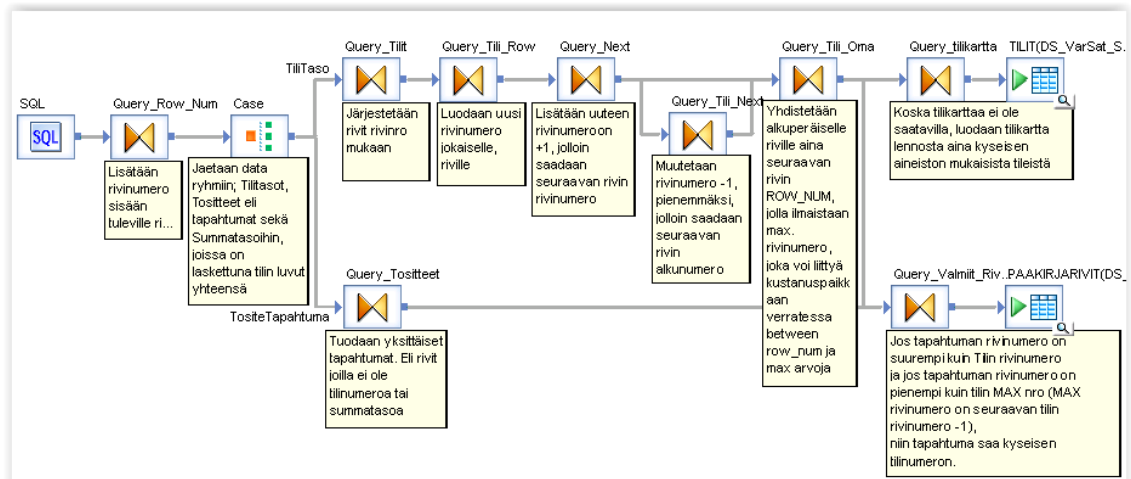
Lähdetiedostoiden tietokantaan latauksessa määriteltiin ensin tiedoston sijaintipolku, tiedoston nimi ja tarvittaessa välilehden nimi. Tiedoston sisältö luetaan BODS:iin, jolloin siitä tulee Excel tiedosto (Excel formaateista) tai flat file (csv formaateista). Tämän jälkeen tiedostosta luetaan tiedot tietokantaan yksinkertaisella läpiviennillä (Kuva 15.), jossa koodistojen tietorakenteeseen ei tehty muutoksia. Pääkirjan tiedot olivat joka toisella rivillä ja tässä vaiheessa aineistosta poistettiin tyhjät rivit. Rivien poisto tapahtui pääkirjan kantaan tuonnin yhteydessä Query-objektin where -ehtolauseessa.



Kuva 15. Lähdetiedostojen tietokantaan tuonti

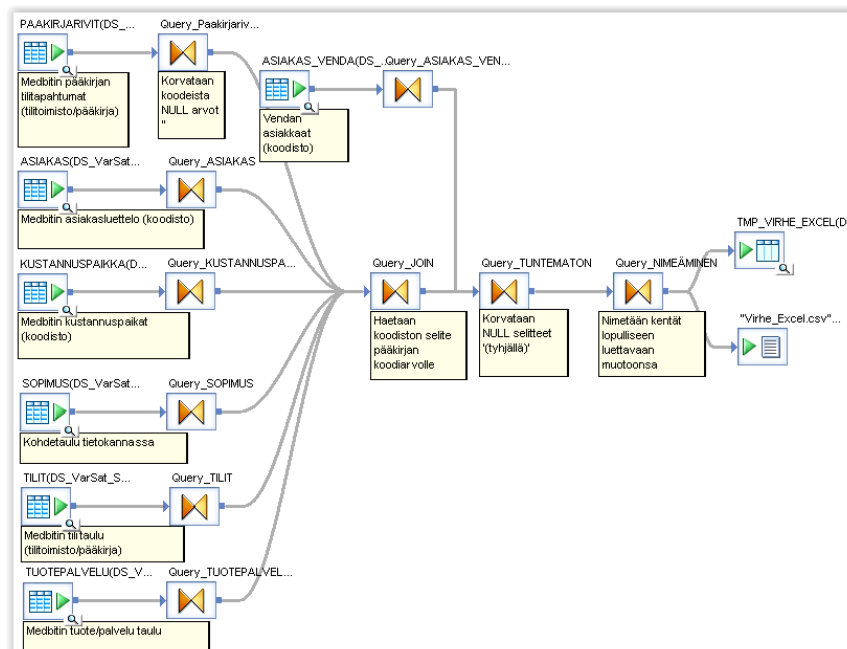
Pääkirja luettiin ensiksi apu_pääkirja-tauluun, koska sille tehtiin muutoksia ja lisäyksiä ennen varsinaiseen tauluun viemistä ja virhe-Excelin luomista. Pääkirjan eurojen tuhatmerkkienä oli ' □ -merkki, joka näkyy Excelissä pitkänä välilyöntinä. BODS ei tunnistanut kyseistä merkkiä, joten eurojen kohdalla tehdyt muutokset toimineet. Välimerkki poistettiin lukemalla apu_pääkirjataulusta SQL-objektilla (Kuva 16.), jolloin muutokset tulivat näkyviin.

Pääkirjan tapahtumariveille lisättiin juokseva numerointi NRO kenttä, joka toimii tapahtumarivin yksilöivänä tunnisteena, koska sellaista ei saatu suoraan tilitoimiston järjestelmästä pääkirjalle mukaan. Numeroinnin lisäyksen jälkeen data jaettiin kahteen osaan (Kuva 16.) Case-objekti. Rivit, joilla on arvo tilinumerosarakkeessa, menivät ylempään putkeen, josta luotiin tilikartta tietokantaan. Rivit, joilla on tositenumero omassa sarakkeessaan, eli tapahtumarivit, menivät alempaan putkeen ja objektissa Query_Valmiit_Rivit, kullekin tapahtumalle lisättiin sille kuuluva tilinumero.



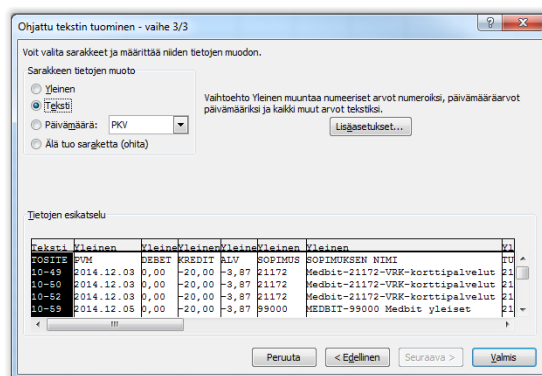
Kuva 16. Tapahtumien yksilöllinen numerointi ja tilikartan luonti

Virhe-Excel koostuu koodistoista, pääkirjasta ja tilikartasta. Objektissa Query_JOIN (Kuva 17.), yhdistetään koodiarvot selitteisiin, jonka jälkeen NULL-arvon saavat kentät korvataan ”tyhjällä”, nvl-funktion avulla. Ennen virhe-Excelin luomista kaikki kentät nimetään vielä luettavaan selkokieleiseen muotoon. Tiedot koostetaan csv-tiedoston lisäksi myös tietokantaan SA-tasolle, väliaikaiseen tauluun.



Kuva 17. Virhe-Excelin luominen

Virhe-Excel luodaan csv-muotoon tositenumeron takia, koska Excelin muotoilusäännöt muuttavat kentän sisällön päivämääräksi. Csv-tiedosto avataan Excelillä teksti lähteenä, jolloin päästään määrittelemään kunkin sarakkeen tietotyytit (Kuva 18.).



Kuva 18. Tekstitiedoston avaaminen Excelillä

Latauksen käynnistyessä ensimmäisenä on komentosarja (Kuva 19.), joka katsoo löytyykö määritellystä polusta tiedosto nimeltä pääkirja. Jos pääkirjaa ei löydy, ajo päättyy siihen ja jos tiedosto löytyy, niin silloin ajo suorittaa kaikki vaiheet läpi virhe-Excelin tulostamiseen asti. Tämän lisäksi on toinen komentosarja, joka siirtää luetun pääkirjan arkistokansioon (Kuva 26.), jolloin kansioden ja tiedostoiden hallinta on helpompaa.

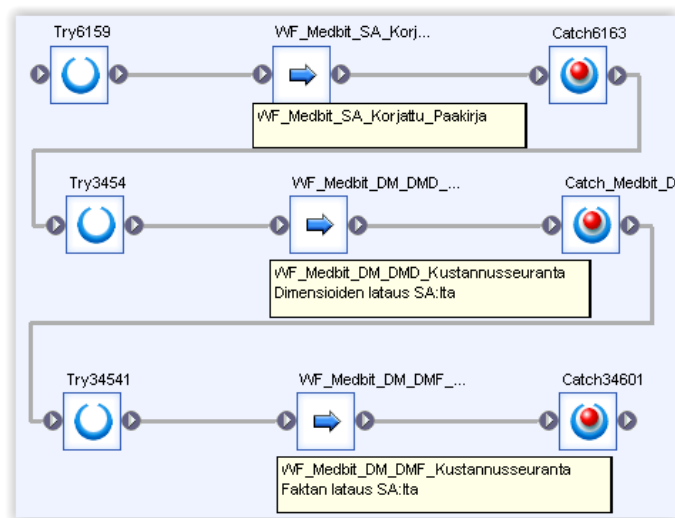
```
#Onko tiedostoa tai tiedostoja lähdekansiossa
#Pääkirjan nimi koostuu vakioidusta nimestä pääkirja*.csv
$G_Ladataanko_paakirja = exec( 'cmd.exe', 'dir/b/a-d [%G_path_in]\pääkirja*.csv | find /v /c ":",2 );
print( 'Tarkistetaan ladataanko SA tasoa vai ei. pääkirja*.csv count tuottaa arvoksi: ' || $G_Ladataanko_paakirja );
```

Kuva 17. Komentosarja joka tarkistaa löytyykö tiedostoa nimeltä pääkirja

4.3 DM-taso

DM-tason lähdetiedostoina toimivat korjattu virhe Excel, sekä SA-tasolla olevien lähdetiedostojen taulut: tili, sopimus, tuote, kustannuspaikka, asiakas, asiakas_Venda ja lopullinen pääkirja.

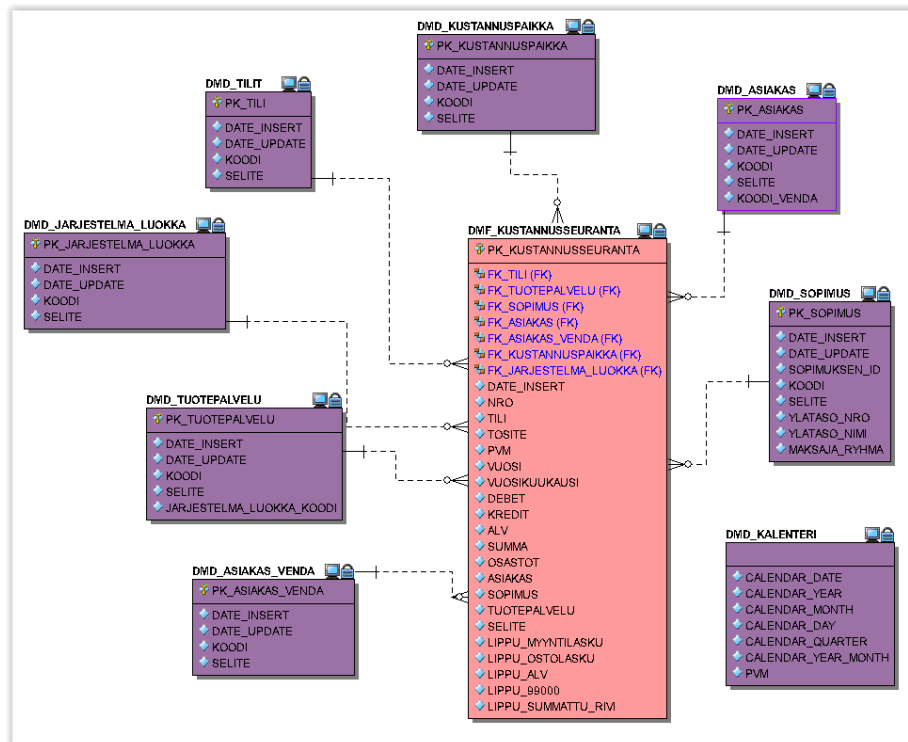
Latauksessa (Kuva 20.) luetaan tietokantaan korjattu virhe-Excel ja lähdetiedot SA-tasolta. Lopullista pääkirjataulua ja virhe-Exceliä verrataan toisiinsa kun virhe-Excelissä olevat korjaukset ladataan tietokantaan DM-tasolle. Tällöin varmistetaan tapahtumarivien yksilöivän tunnuksen kautta, onko manuaalisen korjaustyön aikana jokin rivi vahingossa poistettu. Puuttuvat rivit tuodaan tarvittaessa takaisin SA-tasolta. Tämän jälkeen tiedot siirtyvät lopulliseen muotoonsa dimensio ja fakta tauluihin, joista raportointi voi lukea tiedot.



Kuva 18. DM-tason latauksen kokonaisuuden yleisnäkymä

4.3.1 DM-tason tietokanta

DM-tason tietokanta on suunniteltu dimensionaaliseen tähtimalliin, joka tarkoittaa sitä, että siellä on fakta- ja dimensiotauluja. Kuvassa 21. punainen taulu on faktataulu ja violetit taulut ovat dimensiotauluja.



Kuva 19. ER-studiolla mallinnetut DM-tason taulut

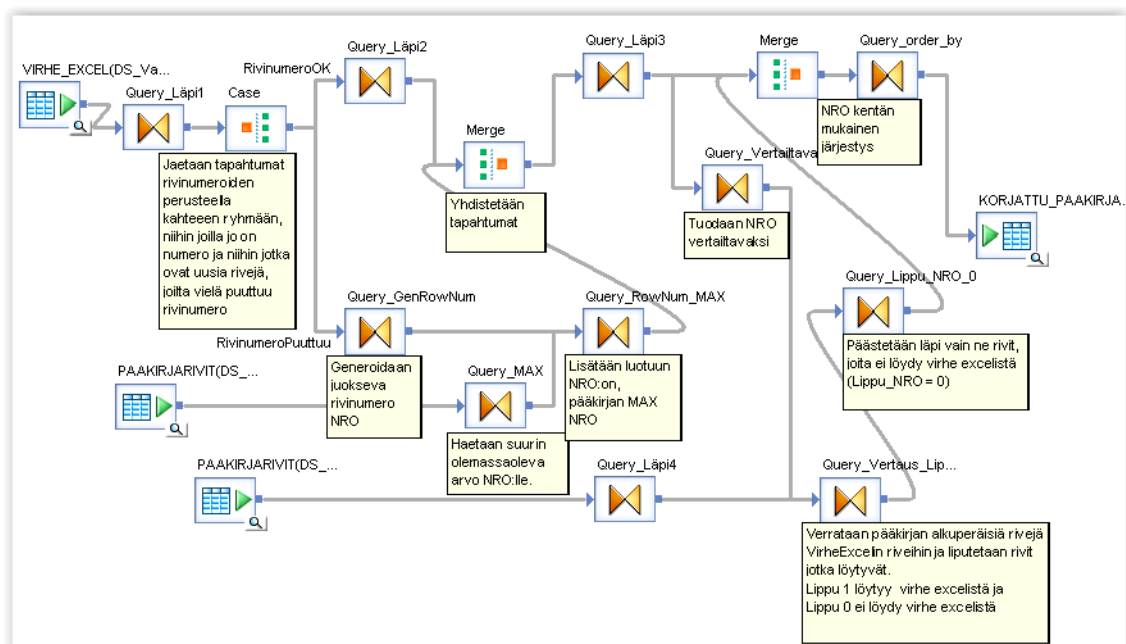
DM-tasolla tauluihin lisättiin Lippu -kentät (Kuva 21.), jotka helpottavat tiedon jaottelussa ja raportoinnissa. Lippu -kentissä arvo voi olla 1 tai 0, jolloin se merkitsee, onko kyseinen rivi jotain vai ei. Käytäntönä on, että kun arvo on 1, se tarkoittaa kyllä ja kun arvo on 0, se tarkoittaa ei. Esimerkiksi rivillä kolme, LIPPU_MYYNTILASKU (Kuva 22.) on arvolla 1, eli se on myyntilasku ja koska sen rivin muut lipputukset ovat arvolla 0, se ei ole mitään muuta. Rivillä viisi, oleva tapahtuma on, sekä ostolasku, että summattu rivi, koska näiden Lippujen arvot ovat 1.

	SOPIMUS	TUOTEPALVELU	LIPPU_MYYNTILASKU	LIPPU_OSTOLASKU	LIPPU_ALV	LIPPU_SOPIMUS_99000	LIPPU_SUMMATTU_RIVI
1	1105	2146	0	0	1	0	0
2	1106	2146	0	0	1	0	0
3	1018	2146	1	0	0	0	0
4	1001	2146	0	1	0	0	0
5	5001	1999	0	1	0	0	1

Kuva 20. Esimerkki Lippu -kentistä

4.3.2 DM-tason Ajologiikka

DM-tason ajo alkaa virhe-Excelin lukemisella tietokantaan, Kuvan 15. mukaisesti. Seuraavaksi rivit jaetaan kahteen putkeen Case-objektissa (kuva 23.), sen perusteella onko NRO-kentässä arvoa. Korjauksen mukana tulleille uusille riveille luodaan myös yksilöivä rivinumero, kohdassa Query_MAX ja Query_RowNum_MAX. Yksilöivä numero lasketaan SA-tason pääkirjan NRO kentän suurimman arvon mukaan. Syy siihen, miksi se haetaan SA-tasolta, eikä virhe-Excelistä on se, että virhe-Excelistä tietojen korjaaja on voinut vahingossa poistaa kyseisen arvon sisältämän rivin. Seuraavat Query-objektit ovat virhe-Excelin vertailua SA-tason pääkirjaan. Tässä kohdassa suoritetaan tarkistus, jolloin katsotaan onko virhe-Excelin korjauskäsittelyn aikana poistettu jokin rivi.

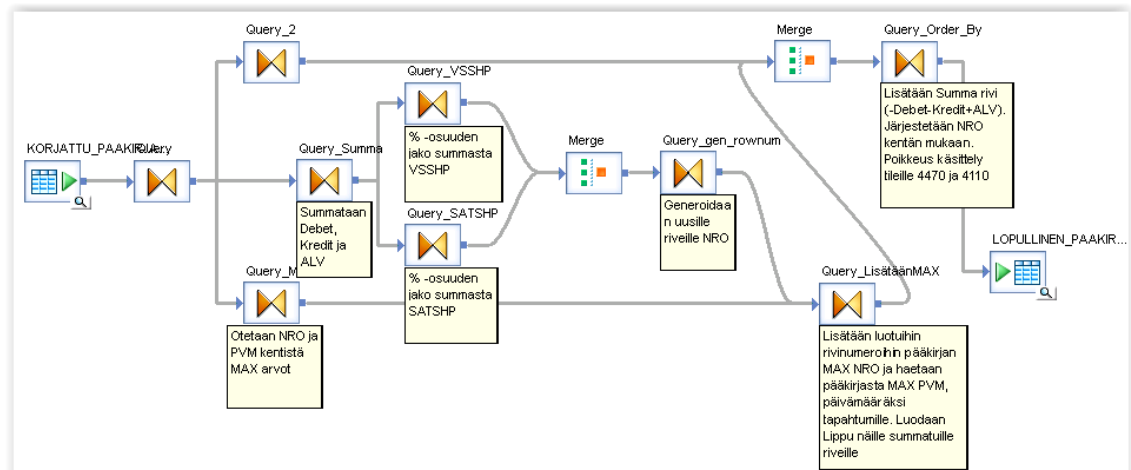


Kuva 21. Korjatun virhe-Excelin lataaminen tietokantaan

Raportointia varten korjattuun pääkirjaan tehtiin vielä laskentasääntöjä ja muutamia poikkeuskäsittelyjä tietyille tileille. Ajossa (Kuva 24.) summataan tietyillä yhteisillä sopimuksilla olevat pääkirjan kustannukset, jotka jaetaan määrätyllä jakoprosentilla sopimuksen asiakkaille. Tästä jaosta tulee pääkirjatauluun kaksi uutta "summa" tapahtumariviä, joille tehtiin lippu Summatturivi = 1, joka erottaa

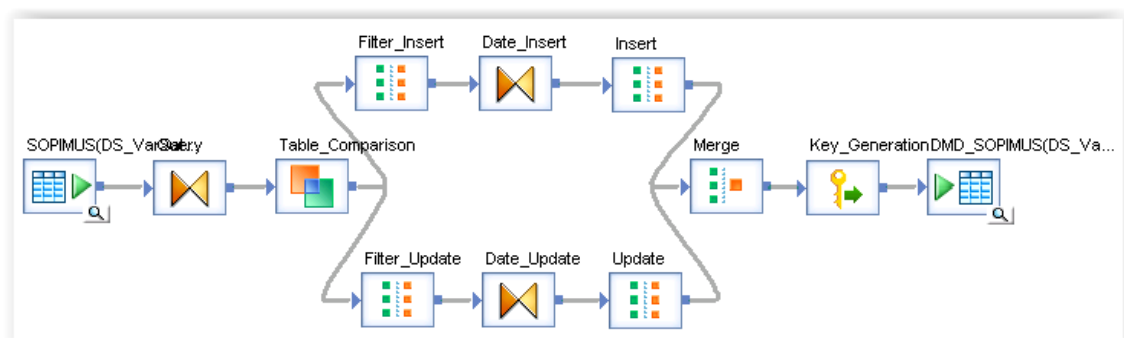
ne muista, oikeista tapahtumariveistä. Tietokantaan laskettiin vielä jokaiselle tapahtumalle tilitoimistolta saadun laskukaavan mukaisesti summarivi, kaavalla: $(-debet - kredit + ALV)$.

Poikkeuksena oli muutama tili, joiden kohdalla ALV-käsittely tapahtui eritavalla ja nämä huomioitiin ajossa tekemällä näille omat laskusäännöt.



Kuva 22. Pääkirjan laskentasäännöt ja poikkeuskäsittelyt

Koodistojen tuonti SA-tasolta DM-tasolle vaatii taulujen vertailun uusille ja muutuneille riveille (Kuva 25.), koska jo korjatulle virhe-Excelille voi tulla lisä korjauksia tietokantaan tuomisen jälkeenkin.



Kuva 23. Koodistojen tuonti SA-tasolta DM-tasolle

Tässä kohdassa ajoa sisältö jaetaan kahteen putkeen, ylämpään menevät uudet rivit, jotka saavat aikaleiman kantaan tuontihetken mukaan ja alempaan menevät vanhat rivit, joissa on tapahtunut jokin muutos. Esimerkiksi tapahtu-

man sopimus on voinut muuttua, tällöin rivi saa aikaleiman päivityshetken mukaan (Kuva 26.). Key_generation-objekti luo riveille PK_SOPIMUS arvon (Kuva 26.), joka toimii DM-tasolla yksilöivänä arvona.

	PK_SOPIMUS	DATE_INSERT	DATE_UPDATE
208	207	2014-12-22 13:01:44....	NULL
209	208	2014-12-22 13:01:44....	NULL
210	209	2014-12-22 13:01:44....	NULL
211	210	2014-12-22 13:01:44....	NULL
212	211	2014-12-22 13:01:44....	NULL

Kuva 24. Insert- ja Update-objektit sekä Key Generator

DM-tason ajon käynnistyessä on vastaavanlainen komentosarja kuin SA-tason latauksessa (Kuva 19.), joka tarkistaa ensimmäisenä löytyykö korjattua pääkirjaa määrätystä kansioista, jos korjattu pääkirja löytyy, ajo suoritetaan ja jos sellaista ei löydy ajo loppuu siihen. Tämän lisäksi latauksesta löytyy komentosarja (Kuva 27.), joka siirtää luetun pääkirjan pääkansioista, arkistokansioon ja nimeää sen uudelleen muotoon Pääkirja_siirtopäivämäärä.

```
# Lataus on onnistunut joten siirretään tiedostot Arkisto kansioon
#exec('cmd.exe', 'MOVE [%G_path_virhe_in]\pääkirja*korjattu.csv [%G_path_arkisto_valmis]', 0):
exec('cmd.exe', 'MOVE [%G_path_virhe_in]\pääkirja*korjattu.csv [%G_path_arkisto_valmis]\pääkirja_korjattu_[%G_tiedoston_pvm].csv', 0):

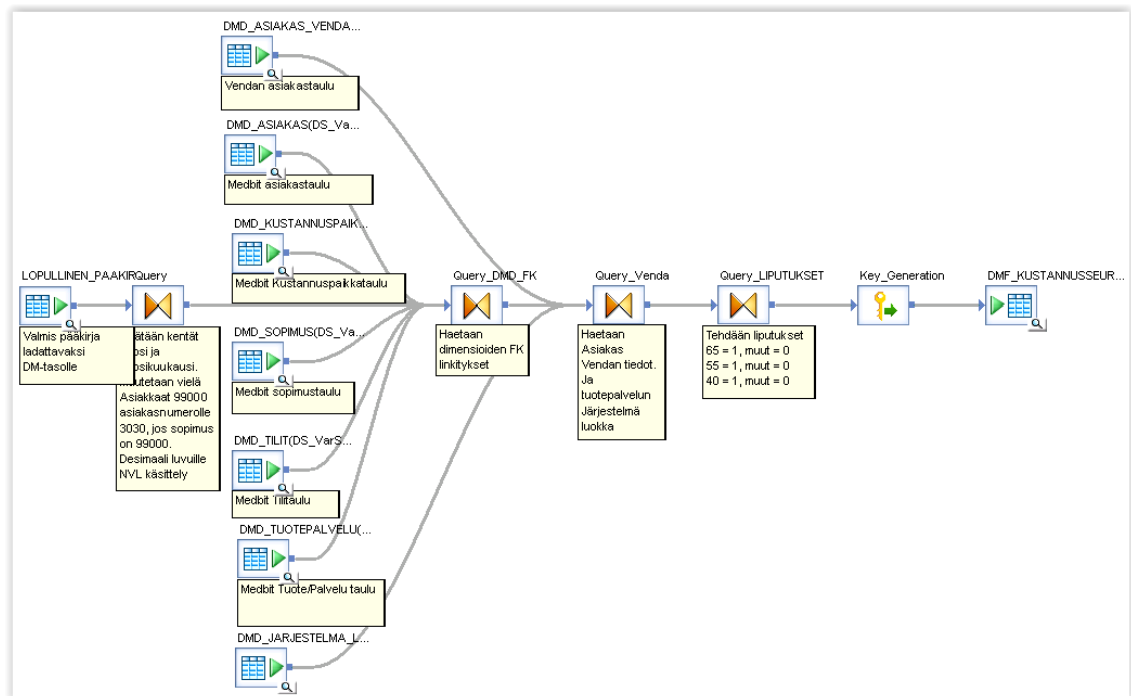
# Loppuun vielä printti että homma on tullut valmiiksi.
print('Pääkirja tiedosto ladattu ja siirretty.')
```

Kuva 25. Komentosarja tiedoston lukemisen jälkeen arkistoon siirtoa varten

DM-tason faktataulu koostuu korjatusta ja täydennetystä pääkirjasta, sekä koodistojen dimensiotauluista (Kuva 28.). Tässä vaiheessa tehtiin liputukset, joissa eri tositelajit erotellaan toisistaan (Kuvat 22. ja 28.). Liputus tapahtuu yksinkertaisella funktiolla, *ifthenelse((Query_Venda.TOSITE like '65%'), 1, 0)*, eli jos edellisestä kyselystä tulevan tositteen arvo alkaa 65-, lippu saa arvoksi 1, muussa tapauksessa arvoksi tulee 0. Eri tositelajeja ovat osto- ja myyntilaskut sekä muistiotositteet ja ALV-tositteet.

Tämän lisäksi faktataulun latauksessa tuntemattomaksi jäävät koodiarvot saavat arvoksi -1, ja tuntemattomaksi jäävät selitteet saavat selitykseksi tuntema-

ton. Tämä tehdään siksi, että raportilla ei haluta näyttää loppukäyttäjälle enää NULL-arvoja.



Kuva 26. Faktataulun lataus

Pääkirjan aineistoon voidaan tarvittaessa tehdä korjauksia vielä virhe Excelin sisäänajon jälkeenkin, jolloin se joudutaan ajamaan uudelleen tietokantaan. Tässä vaiheessa ei ole järkevää yrittää noukkia korjatusta aineistosta yksittäisiä rivejä, joten koko aineisto ajetaan sisään uudelleen. Tätä varten tehtiin komentosarja (Kuva 29.), joka putsaa faktataulusta rivit virhe-Excelissä olevan vuosikuukauden perusteella, jotta uudelleen sisään ajettava aineisto ei olisi faktataulussa kahteen kertaan.

```
#Tarkistetaan ja puhdistetaan faktasta rivit, jotka ovat tulossa uudestaan SA-tasolta.
#Tällä varmistetaan että sama aineisto ei ole kahta kertaa

#Kehityk
sql('DS_VarSat_DM_Medbit','delete FROM DMF_KUSTANNUSSEURANTA where vuosikuukausi in (SELECT distinct convert(varchar(6),pvm,112) FROM Varsat_SA_Medbit.dbo.lopullinen_pankirja)');

#Testaus
sql('DS_VarSat_DM_Medbit','delete FROM DMF_KUSTANNUSSEURANTA where vuosikuukausi in (SELECT distinct convert(varchar(6),pvm,112) FROM Medbit_SA_Testi.dbo.lopullinen_pankirja)');

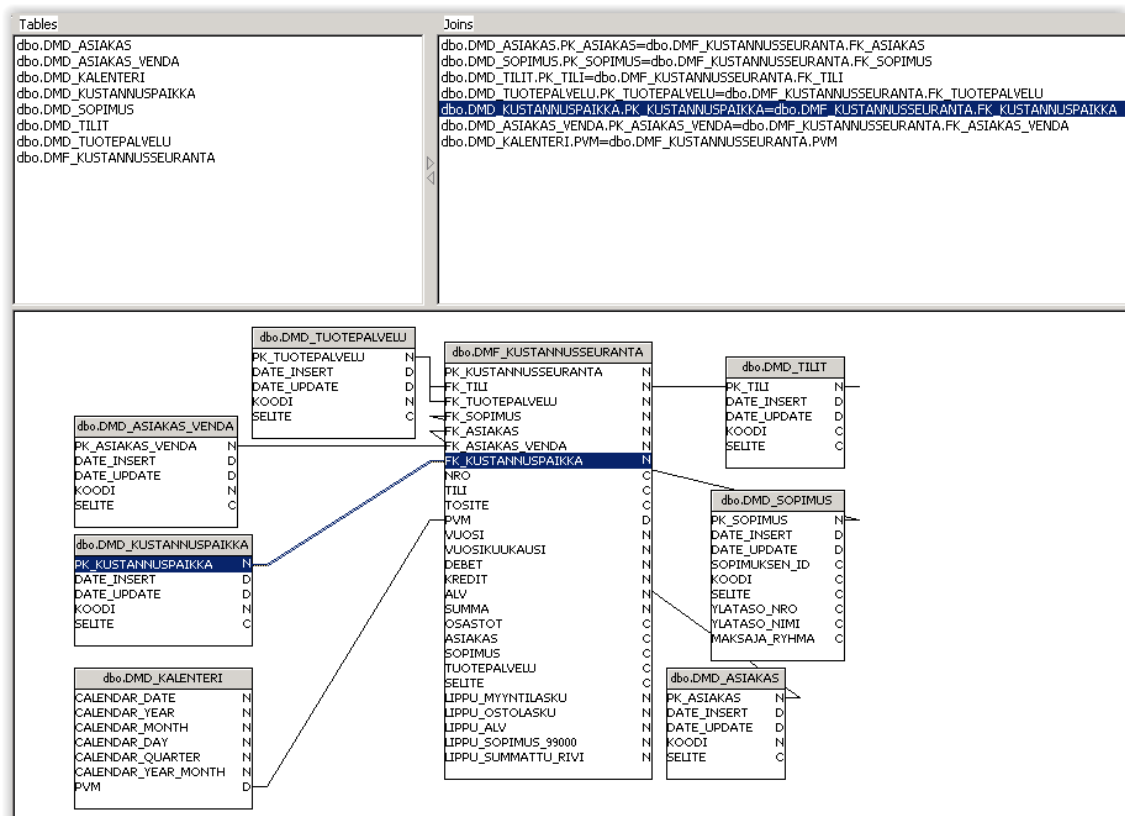
#Tuotanto
sql('DS_VarSat_DM_Medbit','delete FROM DMF_KUSTANNUSSEURANTA where vuosikuukausi in (SELECT distinct convert(varchar(6),pvm,112) FROM Medbit_SA.dbo.lopullinen_pankirja)');
```

Kuva 27. Komentosarja joka tyhjentää DM-tason faktataulun

5 UNIVERSE JA RAPORTOINTI

5.1 Universe

Universe rakennettiin tietokannan DM-tason päälle. Universen tekeminen aloitettiin luomalla kantayhteydet tietokantaan, jonka jälkeen dimensiotaulut linkitettiin faktatauluun (Kuva 30.). Esimerkiksi kustannuspaikka dimensiosta linkitetään PK_Kustannuspaikka -kenttä faktataulun FK_Kustannuspaikka -kenttään. Eli kustannuspaikan dimensiotaulun pääavain linkitetään faktataulussa olevan kustannuspaikan vierasavaimeen.

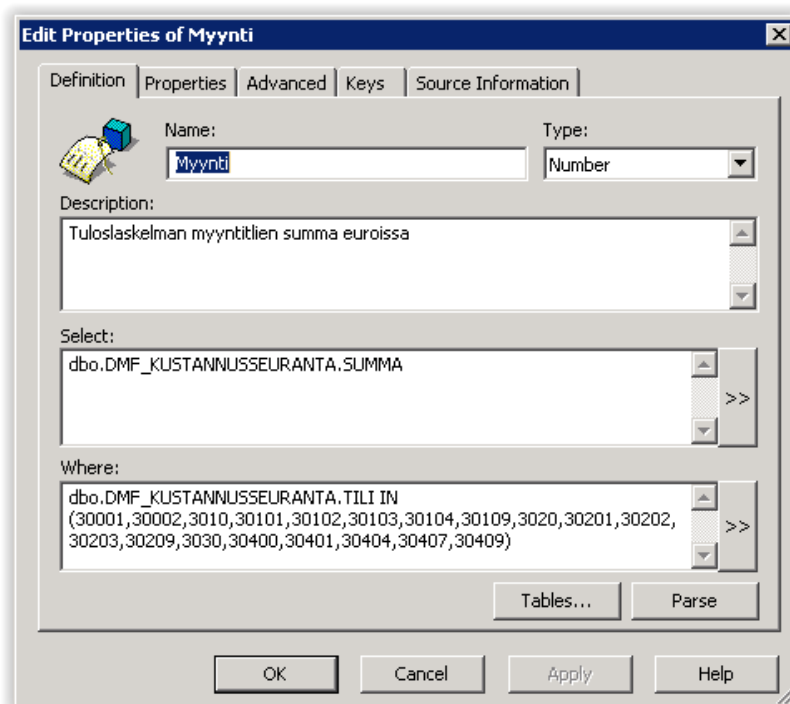


Kuva 28. Universe-kerroksen kaavio, dimensio- ja faktataulujen välisen linkityksen luominen

Linkityksien jälkeen luotiin tarvittavat objektit ja mittarit (Kuva 31.). Jokaiselle objektille ja mittarille määriteltiin tietotyypit ja kentät nimettiin sellaisiksi kuin ne halutaan näkyvän loppukäyttäjille. Objektien ja mittareiden tekemisessä rajauk-

set määritellään SQL-kielestä tutuilla komennoilla select ja where. Select-lauseella valitaan mittariin tai objektiin haluttu kenttä ja where-lauseella hakua voidaan rajata halutusti. Esimerkiksi Myynti -mittarin where-lauseessa otettiin mukaan vain tietyt tilit, jotka kuuluvat myyntitileihin (Kuva 31.).

Universeen voidaan luoda täysin uusia mittareita, joita ei löydy tietovarastosta, eli ne luodaan Universe kerrokseen vain raportointia varten. Myynti -mittari on esimerkki tällaisesta, siinä lasketaan yhteen kaikkien myyntitilien summa niin, että kannassa olevasta summataulusta otetaan mukaan vain tietyillä määritellyillä tilinumeroilla varustetut rivit. Universeen luotiin uudet mittarit: Liikevaihto, joka tulee mittareista ”Myynti – Menot”, ja Menot, jossa on mukana kaikki kulutit. Lista universen objekteista ja mittareista näkyy kuvassa 34. vasemmassa laidassa, kyseinen kuva on Web Intelligence-raportoinnin puolelta.



Kuva 29. Mittarin luominen Universe -kerrokseen

5.2 Web Intelligence-raportointi

Raportointi perustuu Universe-kerroksessa oleviin mittareihin ja dimensioihin. Kustannusseurantaa varten tehtiin kolme erillistä raporttia, yksi jossa on kaikki tapahtumat ja tapahtumien kaikki tiedot (Kuva 32.). Tälle raportille laskettiin kaikkien tapahtumien kustannuksille summat (debet, kredit ja ALV), sekä tili-kaudenvoitto ja myynnit ja kulut. Nämä tiedot päivittyvät sen mukaan, millaisia rajoituksia käyttäjä valitsee katsellessaan raporttia. Raportin ylälaudassa näkyvät kuukausi ja vuosi tiedot päivittyvät automaattisesti, sen mukaan mitkä kuukaudet ja vuodet raportti sisältää.

Medbit - Kustannuseuranta 1 - 12 / 2 014

Raportin viimeisin päivityshetki 9.2.2015 15.51

Sivu/Sheet 1/57

	DEBET	KREDIT	ALV	TILIKAUDENVOITTO (TAPPO)	Myyntit	Kulut
SUMMA	32 878 805,49	-45 794 367,37	-2 662 588,32	10 207 450,71	29 333 627,61	-19 126 176,9

Maksajryhmä	Asiakas - Selite	Sopimus - Selite	Tuotepalvelu - Selite	Tili - Selite	P/V/M	Debet
		Medbit-1-18 Palvelunkapasiteetti teknisen huollon sovelluksia vasten	Palvelunkapasiteettipalvelu	Myynti 24% Jatkuva	6.3.2014	
		MEDBIT-21022-2013 VRK	VRK-korttipalvelu	Myynti 24% Jatkuva	28.2.2014	
		MEDBIT-21022-2013 VRK	VRK-korttipalvelu	Myynti 24% Jatkuva	30.4.2014	
		MEDBIT-21022-2013 VRK	VRK-korttipalvelu	Myynti 24% Jatkuva	30.6.2014	
		Medbit-21022 AIMI	AIMI	Myynti 24% Alkijärje	17.1.2014	

Refresh Date: 9. helmikuuta 2015 klo 15.51.25

Kuva 30. Web Intelligence-raportin kaikki tapahtumat

Kaksi muuta raporttia ovat keskenään samantyylliset, toisessa näkyy sopimukset (Kuva 33.), joiden alla on kullekin sopimukselle kuuluvat palvelut. Toisessa on palvelut ja kunkin palvelun alla näkyy kaikki sopimukset, joille kyseinen palvelu liittyy. Kullekin raportille tehtiin tarvittavat suodatusvaihtoehdot, jotka näkyvät raportin yläpuolella (Kuvat 32. ja 33).

BUSINESSOBJECTS INFOVIEW

Home | Document List | Open | Send To | Dashboards

Web Intelligence - Kustannusseuranta 2014

Document | View | 100% | 1

Sopimus - ID (All values) | Sopimus - Yritys - nimi (All values) | Maksajaryhmä (All values) | Asiakas - Selite (All values) | Kalenteri - Kuukausi (All values)

Medbit-10-2 Sähköinen asiointi

Tuotepalvelu - Selite	Debet	Kredit	ALV	Rivisumma
Sähköinen asiointi	0	-24 552	-4 752	19 800
Sum:	0	-24 552	-4 752	19 800

Medbit-10-3 Alueellinen silmänpohjakuvauspalvelu

Tuotepalvelu - Selite	Debet	Kredit	ALV	Rivisumma
Silmänpohjan kuvauspalvelu	2 400	-12 895,68	-2 263,68	8 232
Sum:	2 400	-12 895,68	-2 263,68	8 232

Medbit-10-5 eResepti

Tuotepalvelu - Selite	Debet	Kredit	ALV	Rivisumma
eResepti / Sähköisen lääkemääräyksen välitys	0	-9 151,2	-1 771,2	7 380
Sum:	0	-9 151,2	-1 771,2	7 380

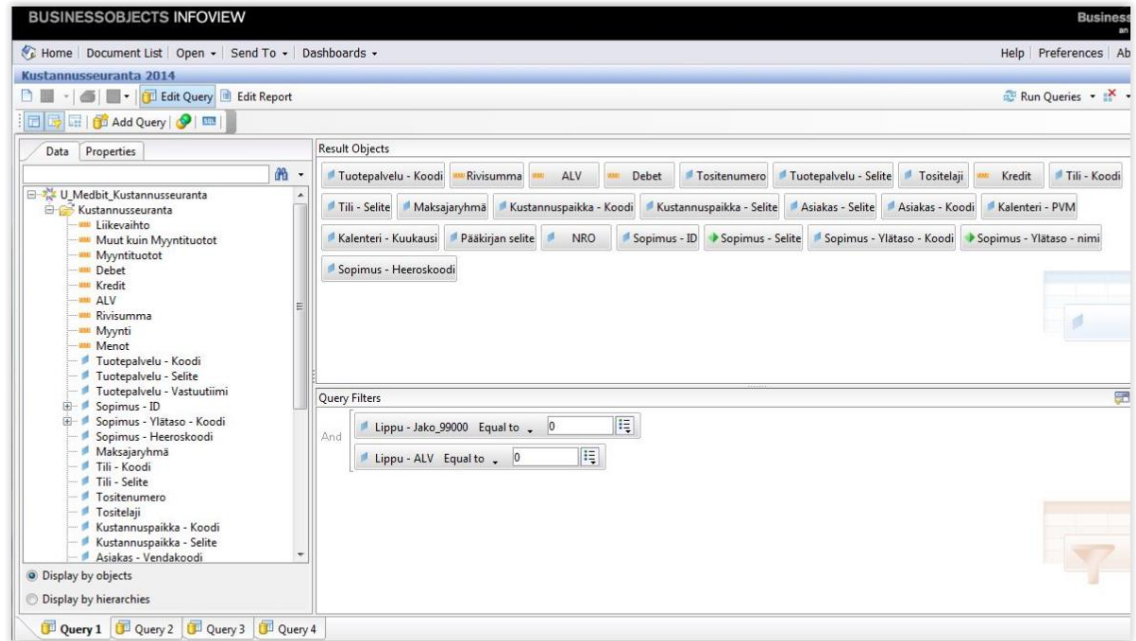
Medbit-10-6-2013 Altti

Tuotepalvelu - Selite	Debet	Kredit	ALV	Rivisumma
Altti	0	-6 249,6	-1 209,6	5 040
Sum:	0	-6 249,6	-1 209,6	5 040

Tapahtumat | Sopimukset | Tuotepalvelu

Kuva 31. Web Intelligence-raportti, sopimukset välilehti

Raportti voidaan luoda tekemällä yksi tai useampi kysely (Query), johon valitaan raportille tarvittavat kentät ja määritellään rajausehdot. Kustannusseurannan raportteja varten luotiin neljä eri kyselyä, jotta raportit saatiin toimimaan halutulla tavalla. Ensimmäisessä kyselyssä tuotiin mukaan kaikki tapahtumarivit, joten siitä rajattiin pois aikaisemmin jaetut yhteiset kustannukset, koska nämä eivät ole alkuperäisiä tapahtumarivejä. Jaetut rivit tuotiin mukaan erillisenä toisessa kyselyssä, jolloin raportin suodatukset ja laskukaavat saatiin toimimaan oikein. Kyselyt 3 ja 4 ovat keskenään samanlaiset, nämä kyselyt ovat sopimukset ja palvelu välilehdillä olevia raportteja varten, eikä niihin tarvinnut tuoda kaikkia samoja tietoja, joita ensimmäiselle raportille tuotiin.



Kuva 32. Web Intelligence-raportin kyselyn luominen

6 YHTEENVETO

Opinnäytetyön tarkoituksena oli luoda loppukäyttäjän tarpeita vastaava Web Intelligence-raportointi. Raportoinnin tavoitteena oli parantaa yrityksen omien kustannustenseurantaa. Työssä käydään läpi myös liiketoimintatiedon hallintaa ja tietovaraston teoriaa.

Toimeksiantona suoritettu käytännön osuus saatiin valmiiksi projektisuunnitelman mukaisesti, vaikka aikataulu olikin hyvin tiukka. Käytännönosuus piti sisällään tietokantojen luomisen, ajologiikan tekemisen (ETL -kehityksen), Universen luomisen tietovaraston päälle ja raportoinnin tekemisen. Tämä kaikki toteutettiin kehitysympäristössä, josta siirryttiin projektin valmistuttua tuotantoympäristöön. Tuotantoympäristöön siirtymisen jälkeen tapahtuva kehitys tapahtuu kuitenkin kehitysympäristössä, jossa muutokset todetaan ensin toimiviksi ja vasta sitten siirretään tuotantoympäristöön.

Projektia viivästytti yrityksen ulkopuolisen tahon kanssa asiointi, sekä raportin syntymisen kiertokulussa oleva virhe-Excelin korjausvaihe, joka käsitti koko vuoden aineiston kohdistusvirheiden korjaamisen manuaalisesti.

Kustannusseuranta projektin kehitystä jatketaan Medbitissä nyt rauhallisemmalla aikataululla ja raportointia on tarkoitus laajentaa niin, että kustannuksien vierele saataisiin myös kirjattavat projektityötunnit, jolloin yrityksen sisällä pystyttäisiin seuraamaan tarkemmin ja tehokkaammin projekteihin ja sovelluksien ylläpitoon käytettyä työaikaa ja niiden kustannuksia. Tarkoituksena olisi laajentaa raportointia myös niin, että osastokohtaiset kustannukset saataisiin helpommin näkyviin ja seurantaan. Tällöin myös raportoinnin käyttäjäkunta laajenisi huomattavasti. Organisaatiossa on tulossa kaikkien lähdekoodistojen muutos hierarkiseksi ja helpommin tulkittaviksi, mikä tarkoittaa sitä, että ajologiikkaan, tietokantoihin ja raportointiinkin tulee muutoksia, jotta ne vastaisivat uusia koodistoja.

LÄHTEET

- [1] Hovi, A.; Hervonen, H. & Koistinen, H. 2009. Tietovarastot ja Business Intelligence, WSOY-pro/Docendo.
- [2] Rouse, M. 2014. Business Intelligence (BI). Viitattu 10.4.2015
<http://searchdatamanagement.techtarget.com/definition/business-intelligence>
- [3] Ellä, H. 2015. Medbit Oy, henkilökohtainen tiedonanto. Viitattu 22.4.2015
- [4] 2009. ETL process. Viitattu 20.3.2015
<http://datawarehouse4u.info/ETL-process.html>
- [5] 2015. ETL (Extract-Transform-Load). Viitattu 8.4.2015
<http://www.dataintegration.info/etl>
- [6] 2015. Wikipedia. Viitattu 6.4.2015
<http://en.wikipedia.org/wiki/BusinessObjects>
- [7] 2015. SAP. Viitattu 2.4.2015
<http://www.sap.com/corporate-en/about/our-company/>
- [8] Rose, J. & Kuruvilla T. 2015. SAP BusinessObjects BI 4.1 Upgrade. Viitattu 29.4.2015
<https://www.sapappsdevelopmentpartnercenter.com/en/get-started/sap-business-intelligence/>
- [9] Gunasekar, V.K. 2014. Viitattu 27.4.2015
<http://scn.sap.com/docs/DOC-52737>
- [10] 2015. SAP Solution Brief. Viitattu 16.4.2015
https://www.insight.com/content/dam/insight/en_US/pdfs/sap/sap-business-objects-enterprise.pdf
- [11] Standen, J. 2008. Data Warehouse vs Data Mart. Viitattu 14.4.2015
<http://www.datamartist.com/data-warehouse-vs-data-mart>
- [12] 2015. Data Marts. Viitattu 14.4.2015
<https://technet.microsoft.com/en-us/library/aa905978%28v=sql.80%29.aspx>
- [13] Rouse, M. 2014. Data Mart. Viitattu 15.4.2015
<http://searchsqlserver.techtarget.com/definition/data-mart>
- [14] Standen, J. 2008. Dimensional Tables and Fact Tables. Viitattu 14.4.2015
<http://www.datamartist.com/dimensional-tables-and-fact-tables>
- [15] 2015. Tietokannan normalisointi. Viitattu 17.4.2015
http://fi.wikipedia.org/wiki/Tietokannan_normalisointi